

Big Data GHW – 3

Name: Sukriti Macker

NETID: sm11017

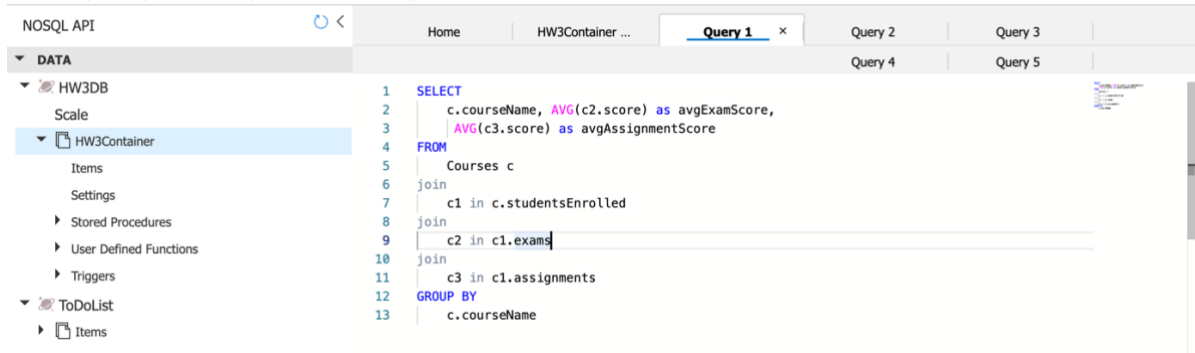
Course: Big Data Section – D

Semester: Fall 2023

Q1. Fetch the course name and the average exam and average assignment score for all courses.

Solution 1.

For the above question, the query executed is:-



The screenshot shows a database query editor with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query for 'Query 1'. The query is as follows:

```
1 SELECT
2   c.courseName, AVG(c2.score) as avgExamScore,
3   AVG(c3.score) as avgAssignmentScore
4 FROM
5   Courses c
6   join
7   c1 in c.studentsEnrolled
8   join
9   c2 in c1.exams
10  join
11  c3 in c1.assignments
12 GROUP BY
13  c.courseName
```

The output of the query is:-



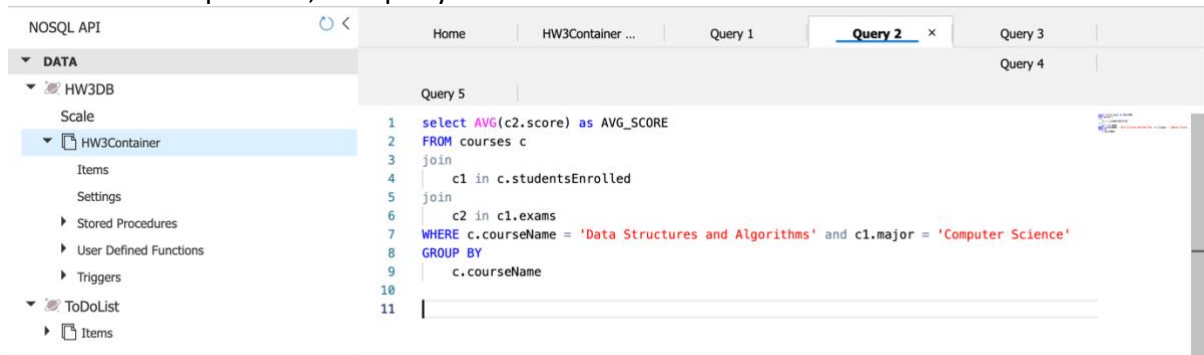
The screenshot shows the 'Results' tab of a database query results viewer. It displays a JSON array of five objects, each representing a course and its average exam and assignment scores. The output is as follows:

```
[
  {
    "courseName": "Advanced DBMS",
    "avgExamScore": 88,
    "avgAssignmentScore": 84
  },
  {
    "courseName": "Intro to Machine Learning",
    "avgExamScore": 90.25,
    "avgAssignmentScore": 90.5
  },
  {
    "courseName": "Network Security",
    "avgExamScore": 81.25,
    "avgAssignmentScore": 79
  },
  {
    "courseName": "Operating System",
    "avgExamScore": 89,
    "avgAssignmentScore": 91.25
  },
  {
    "courseName": "Data Structures and Algorithms",
    "avgExamScore": 90.25,
    "avgAssignmentScore": 90.75
  }
]
```

Q2. Calculate the average exam score for students majoring in "Computer Science" who are enrolled in the "Data Structures and Algorithms" course.

Solution 2.

For the above question, the query executed is:-



The screenshot shows the NOSQL API interface. On the left, a sidebar lists the database structure: DATA, HW3DB, Scale, HW3Container (selected), Items, Settings, Stored Procedures, User Defined Functions, Triggers, and ToDoList. The main area displays the SQL query for Query 2:

```
1 select AVG(c2.score) as AVG_SCORE
2 FROM courses c
3 join
4   c1 in c.studentsEnrolled
5 join
6   c2 in c1.exams
7 WHERE c.courseName = 'Data Structures and Algorithms' and c1.major = 'Computer Science'
8 GROUP BY
9   c.courseName
10
11
```

The output of this query is:-



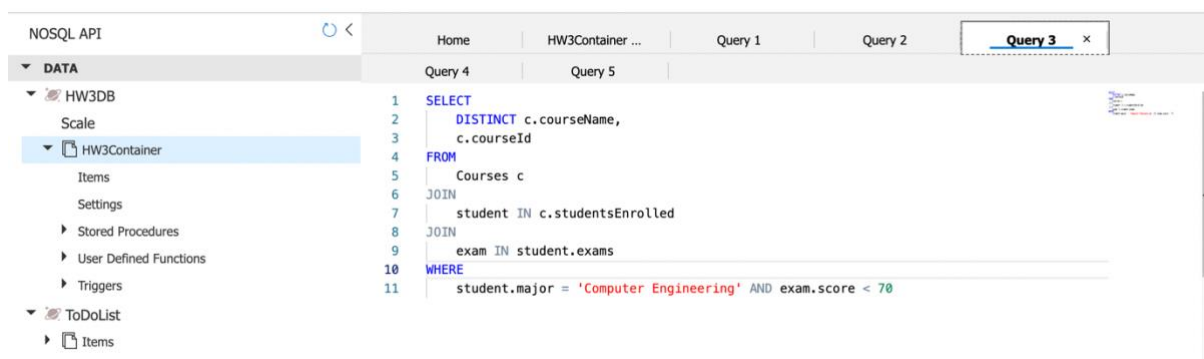
The screenshot shows the NOSQL API interface with the 'Results' tab selected. It displays the output of the query for Query 2, which is a JSON object:

```
1 - 1
[
  {
    "AVG_SCORE": 88.5
  }
]
```

Q3. Find the courses where at least one student's major is "Computer Engineering" and they scored below 70 on exams.

Solution 3.

For the above question, the query executed is:-



The screenshot shows the NOSQL API interface. On the left, a sidebar lists the database structure: DATA, HW3DB, Scale, HW3Container (selected), Items, Settings, Stored Procedures, User Defined Functions, Triggers, and ToDoList. The main area displays the SQL query for Query 3:

```
1 SELECT
2   DISTINCT c.courseName,
3   c.courseId
4 FROM
5   Courses c
6 JOIN
7   student IN c.studentsEnrolled
8 JOIN
9   exam IN student.exams
10 WHERE
11   student.major = 'Computer Engineering' AND exam.score < 70
```

The output of the query:-

Results Query Stats

1 - 1

```
{
  "courseName": "Network Security",
  "courseId": "104"
}
```

Q4. Select any two courses where there is at least one student's major that is "Computer Science" (There must be at least three students in different courses, who's major is Computer Science)

Solution 4.

For the above question, the query executed is:-

NOSQL API

DATA

- HW3DB
 - Scale
 - HW3Container
 - Items
 - Settings
 - Stored Procedures
 - User Defined Functions
 - Triggers
- ToDoList
 - Items

Query 4

```
1 SELECT
2   c.courseName,
3   c.courseId
4 FROM
5   Courses c
6 JOIN
7   student IN c.studentsEnrolled
8 WHERE
9   student.major = 'Computer Science'
10 OFFSET 0 LIMIT 2
```

The output of the query:-

Results Query Stats

1 - 2

```
{
  "courseName": "Data Structures and Algorithms",
  "courseId": "102"
},
{
  "courseName": "Operating System",
  "courseId": "103"
}
```

Q5. Calculate the overall pass grade (if they passed or failed) for each student, considering a passing grade as an exam score (exams) of 70 or higher. (Make sure at least one student fails in your database)

Solution 5.

For the above question, the query executed is:-

```

1 SELECT
2   c.courseName,
3   student.studentName,
4   student.major,
5   ARRAY(
6     SELECT
7       exam.examName,
8       exam.score,
9       (exam.score >= 70) AS isPassed
10    FROM
11      exam IN student.exams
12   ) AS
13   examResults
14 FROM
15   Courses c
16 JOIN
17   student IN c.studentsEnrolled

```

The output of the query:-

```

{
  "courseName": "Data Structures and Algorithms",
  "studentName": "John Doe",
  "major": "Computer Science",
  "examResults": [
    {
      "examName": "Midterm Exam",
      "score": 92,
      "isPassed": true
    },
    {
      "examName": "Final Exam",
      "score": 85,
      "isPassed": true
    }
  ]
}

```

```

{
  "courseName": "Data Structures and Algorithms",
  "studentName": "Ella Brown",
  "major": "Computer Engineering",
  "examResults": [
    {
      "examName": "Midterm Exam",
      "score": 90,
      "isPassed": true
    },
    {
      "examName": "Final Exam",
      "score": 94,
      "isPassed": true
    }
  ]
},
{
  "courseName": "Operating System",

```



```
{
  "courseName": "Network Security",
  "studentName": "Chris Friedman",
  "major": "Computer Engineering",
  "examResults": [
    {
      "examName": "Midterm Exam",
      "score": 80,
      "isPassed": true
    },
    {
      "examName": "Final Exam",
      "score": 68,
      "isPassed": false
    }
  ]
},
{
  ...
}
```

Learning Outcomes:-

- Proficiency in running queries in Cosmos DB.
- Competency in creating and managing data items.
- Understanding NoSQL database principles.
- Familiarity with data modelling in a NoSQL context.
- Gained practical experience in working with a real-world NoSQL database system.