Report on Query Results and Performance Observations

Query Results

1. Students Enrolled in Course (`courseId: 1`)

Spark Query:

```
# Filter students enrolled in courseId 1

enrolled_students =
students_df.filter(students_df.enrollments.courseId ==
1).select("firstName", "lastName", "email", "mobile")



# Show the results

enrolled_students.show()
```

Results:

firstName	lastName	email	mobile
Alice	Johnson	alice@university.edu	1234567890
Bob	Brown	bob@university.edu	0987654321

Charlie	Davis	charlie@university.edu	2345678901

Observation: The query successfully retrieved the details of three students enrolled in the specified course, demonstrating efficient filtering capabilities.

2. Average Enrollment Count for Instructor (`instructorId: 2`)

Spark Query:

```
# Group courses by instructorId and calculate average enrollments

avg_enrollment = courses_df.filter(courses_df.instructorId == 2) \

    .withColumn("enrollmentCount", F.size(F.col("enrollments"))) \


.groupBy("instructorId").agg(F.avg("enrollmentCount").alias("avgEnroll
ment"))



# Show the result

avg_enrollment.show()
```

Results:

instructorId	avgEnrollment
2	1.0

Observation: The average enrollment count for the instructor is 1. This indicates limited course participation, possibly requiring further investigation into course engagement.

3. Courses Offered by Department (`departmentId: 3`)

Spark Query:

```
# Filter courses by departmentId and select courseName and
instructorId

department_courses = courses_df.filter(courses_df.departmentId ==
3).select("courseName", "instructorId")



# Show the result

department_courses.show()
```

Results:

courseName	instructorId
Thermodynamics	4
Heat Transfer	6
Manufacturing Processes	3
Fluid Mechanics	6
Structural Mechanics	3

Material Science	3
Robotics	6

Observation: This query provided a comprehensive list of courses within the Mechanical Engineering department, highlighting course offerings and their respective instructors.

4. Total Number of Students per Department

Spark Query:

```
total_students =
students_df.groupBy("departmentId").agg(F.count("*").alias("totalStudents"))



# Join with departments to get department names

total_students = total_students.join(departments_df, total_students.departmentId ==
departments_df.departmentId) \

    .select("departmentName", "totalStudents")



total_students.show()
```

Results:

departmentName	**totalStudents**
Computer Science	10

Electrical Engineering	8
Mechanical Engineering	6

Observation: The Computer Science department has the highest enrollment, indicating its popularity. This query effectively aggregated data, showcasing Spark's strength in handling group operations.

5. Instructors Teaching Core Courses in CSE Department

Spark Query:

```
# Filter instructors by departmentId and courses taught
instructors_courses =
instructors_df.filter(instructors_df.departmentId == 1) \
    .withColumn("coreCourses", F.expr("filter(courses, course ->
course.departmentId = 1)"))

# Show the result
instructors_courses.select("firstName", "lastName", "email",
"coreCourses").show(truncate=False)
```

Results:

firstName	lastName	email	coreCourses

John	Doe	jdoe@university.edu	[Intro to Programming, Data Structures, DBMS, Adv Programming...]
Emily	Johnson	ejohnson@university.edu	[Web Development, Operating Systems, Computer Networks]
Sophia	Lee	slee@university.edu	[Network Security, AI]

Observation: This query effectively details instructors in the Computer Science department and the core courses they teach, providing insights into faculty engagement.

6. Top 10 Courses by Enrollment Count

Spark Query:

```
# Calculate enrollment count for each course and sort by descending order

top_courses = courses_df.withColumn("enrollmentCount", F.size(F.col("enrollments"))) \

    .orderBy(F.desc("enrollmentCount")).limit(10)


# Show the result

top_courses.select("courseName", "enrollmentCount").show()
```

Results:

courseName	enrollmentCount

Introduction to Programming	3
Digital Signal Processing	2
Network Security	2
Database Management Systems	2

Observation: The course "Introduction to Programming" leads in enrollment, reflecting its foundational importance in the curriculum.