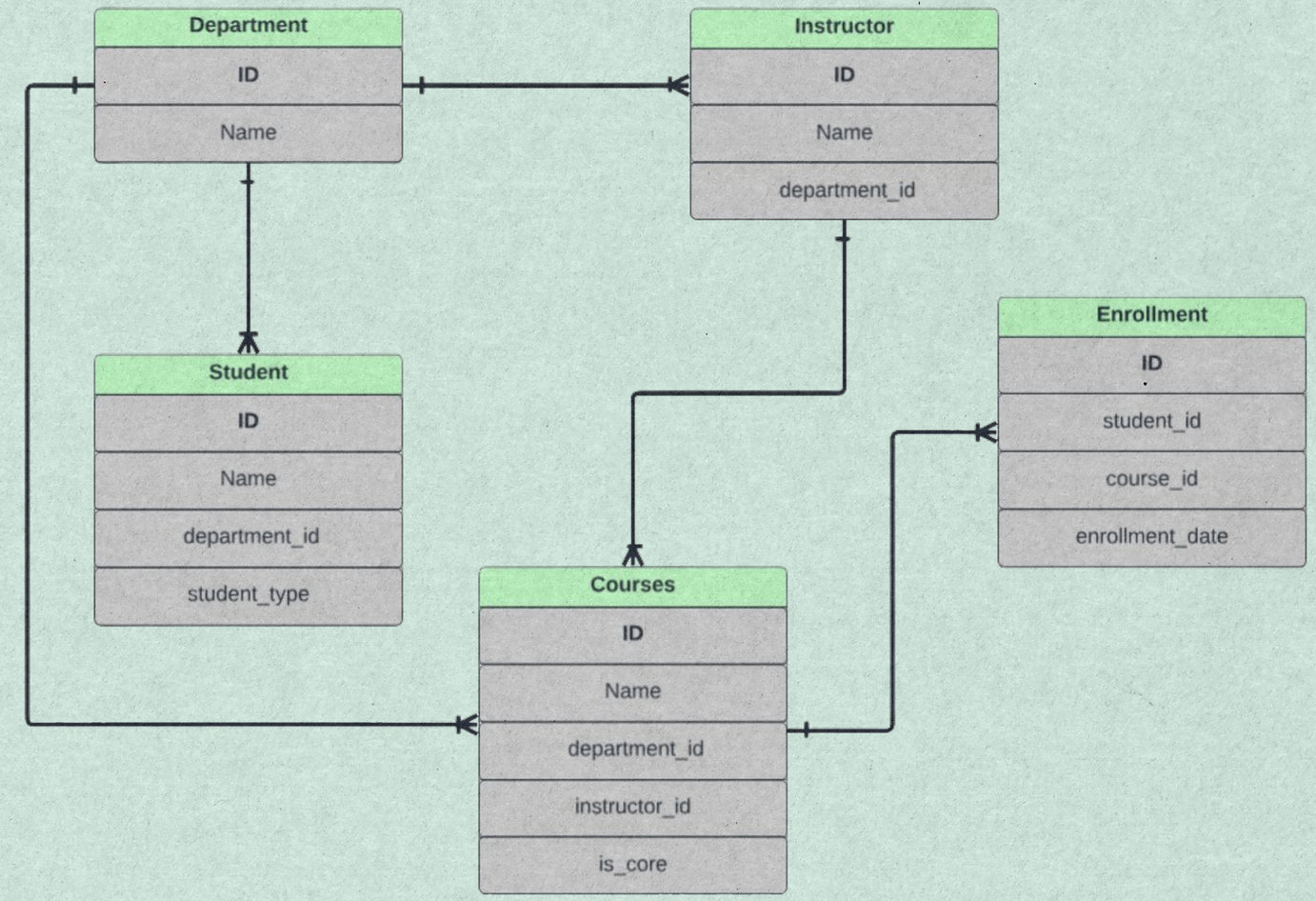
RDBMS to MongoDB migration with Spark

Dikshant Kukreja
2022176

RDBMS SCHEMA

The university database schema consists of five entities: Departments, Instructors, Students, Courses, and Enrollments. Departments link to multiple instructors, students, and courses, while Enrollments capture student-course associations and instructor-course interactions.



RDBMS Queries

Fetch all students in a specific course

```
SELECT s.name, s.student_type
    FROM Students s
    JOIN Enrollments e ON s.id = e.student_id
    JOIN Courses c ON e.course_id = c.id
    WHERE c.name = 'Data Structures';
```

Average students per instructor

```
SELECT AVG(enrollment_count)
    FROM (
        SELECT COUNT(e.student_id) AS enrollment_count
        FROM Enrollments e
        JOIN Courses c ON e.course_id = c.id
        JOIN Instructors i ON c.instructor_id = i.id
        WHERE i.name = 'Ravi Sharma'
        GROUP BY c.id
    ) AS course_counts;
```

List all courses by department

```
SELECT c.name
    FROM Courses c
    JOIN Departments d ON c.department_id = d.id
    WHERE d.name = 'Computer Science';
```

Total number of students per department

```
SELECT d.name AS department_name, COUNT(s.id) AS total_students
    FROM Students s
    JOIN Departments d ON s.department_id = d.id
    GROUP BY d.id;
```

RDBMS Queries

Instructors who taught all BTech CSE core courses

```
SELECT DISTINCT i.name
FROM Instructors i
JOIN Courses c ON i.id = c.instructor_id
WHERE c.is_core = TRUE
AND c.department_id = (SELECT MIN(id) FROM Departments WHERE name = 'Computer Science')
GROUP BY i.id
HAVING COUNT(c.id) = (
    SELECT COUNT(*)
    FROM Courses
    WHERE is_core = TRUE
    AND department_id = (SELECT MIN(id) FROM Departments WHERE name = 'Computer Science')
);
```

Top-10 courses by highest enrollments

```
SELECT c.name, COUNT(e.student_id) AS total_enrollments
FROM Courses c
JOIN Enrollments e ON c.id = e.course_id
GROUP BY c.id
ORDER BY total_enrollments DESC
LIMIT 10;
```

MongoDB Schema

Departments Collection

```
{
  "_id": ObjectId,          // Unique identifier for each department
  "name": String            // Name of the department
}
```

Instructors Collection

```
{
  "_id": ObjectId,          // Unique identifier for each instructor
  "name": String,           // Name of the instructor
  "department_id": ObjectId
}
```

Students Collection

```
{
  "_id": ObjectId,          // Unique identifier for each student
  "name": String,           // Name of the student
  "department_id": ObjectId,
  "student_type": String   // Type of student (BTech, MTech, PhD)
}
```

Courses Collection

```
{
  "_id": ObjectId,     // Unique identifier for each course
  "name": String,      // Name of the course
  "department_id": ObjectId,
  "instructor_id": ObjectId,
  "is_core": Boolean // Indicates if the course is core (true/false)
}
```

Enrollments Collection

```
{
  "_id": ObjectId,            // Unique identifier for each enrollment
  "student_id": ObjectId,     // Reference to the Students collection
  "course_id": ObjectId,      // Reference to the Courses collection
  "enrollment_date": Date     // Date of enrollment
}
```

MongoDB Queries

Fetch all students in a specific course

```
[
    {
        "$lookup": {
            "from": "Enrollments",
            "localField": "_id",
            "foreignField": "student_id",
            "as": "enrollment"
        }
    },
    {
        "$unwind": "$enrollment"
    },
    {
        "$lookup": {
            "from": "Courses",
            "localField": "enrollment.course_id",
            "foreignField": "_id",
            "as": "course"
        }
    },
    {
        "$unwind": "$course"
    },
    {
        "$match": {
            "course.name": "Data Structures"
        }
    },
    {
        "$project": {
            "name": 1,
            "student_type": 1
        }
    }
]
```

Average students per instructor

```
[
    {
        "$lookup": {
            "from": "Enrollments",
            "localField": "_id",
            "foreignField": "course_id",
            "as": "enrollment"
        }
    },
    {
        "$group": {
            "_id": "$instructor_id",
            "enrollment_count": {"$sum": {"$size": {"$ifNull":
["$enrollment", []]}}}
        }
    },
    {
        "$group": {
            "_id": None,
            "average": {"$avg": "$enrollment_count"}
        }
    }
]
```

List all courses by department

```
[
    {
        "$lookup": {
            "from": "Departments",
            "localField": "department_id",
            "foreignField": "_id",
            "as": "department"
        }
    },
    {
        "$unwind": "$department"
    },
    {
        "$match": {
            "department.name": "Computer Science"
        }
    },
    {
        "$project": {
            "name": 1
        }
    }
]
```

MongoDB Queries

Total Number of Students per Department

```
[
    {
        "$lookup": {
            "from": "Departments",
            "localField": "department_id",
            "foreignField": "_id",
            "as": "department"
        }
    },
    {
        "$unwind": "$department"
    },
    {
        "$group": {
            "_id": "department.name",
            "total_students": {"$sum": 1}
        }
    }
]
```

Instructors who taught B.Tech CSE core courses

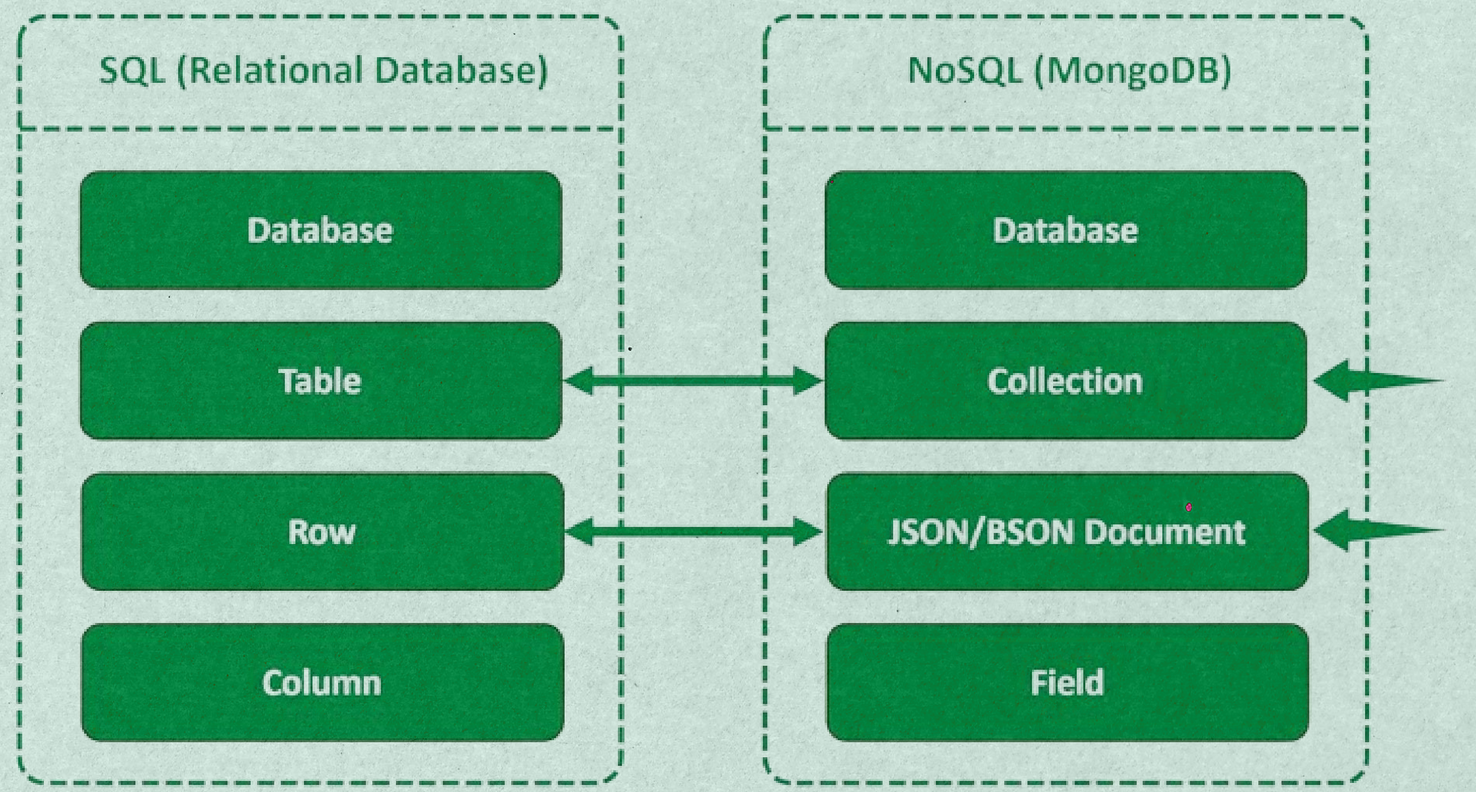
```
[
    {
        "$lookup": {
            "from": "Courses",
            "localField": "_id",
            "foreignField": "instructor_id",
            "as": "courses"
        }
    },
    {
        "$unwind": "$courses"
    },
    {
        "$match": {
            "courses.is_core": True,
            "courses.department_id":
db.Departments.find_one({"name": "Computer Science"})["_id"]
        }
    },
    {
        "$group": {
            "_id": "$_id",
            "count": {"$sum": 1}
        }
    },
    {
        "$match": {
            "count": db.Courses.count_documents({"is_core": True,
"department_id": db.Departments.find_one({"name": "Computer
Science"})["_id"]})
        }
    }
]
```

Top-10 courses by highest enrollments

```
[
    {
        "$lookup": {
            "from": "Enrollments",
            "localField": "_id",
            "foreignField": "course_id",
            "as": "enrollment"
        }
    },
    {
        "$project": {
            "name": 1,
            "total_enrollments": {"$size": {"$ifNull":
["$enrollment", []]}}
        }
    },
    {
        "$sort": {"total_enrollments": -1}
    },
    {
        "$limit": 10
    }
]
```

Data Migration Steps

- For Data Migration i used Psycopg2 and pymongo (mongoclient).
- Both the databases the RDBMS one and the MongoDB one were hosted and created on local servers.
- Data transfer was done post loading in Dataframes and then converting in the desired MonogoDB Schema as mentioned.



Query Implementation and Optimization using Apache Spark

- For implementation, i used Spark context.
- For connecting the local servers, i used Spark sessions by creating a new environment.

- I used the following 2 optimization methods :-
  - Indexing in MongoDB: Creating indexes on frequently queried fields can significantly improve query performance. For instance, you can create indexes on department_id in the Instructors, Students, and Courses collections and on student_id and course_id in the Enrollments collection. This will speed up lookups and joins in your queries.

  - Bulk Inserts: Instead of inserting each document one at a time, you can collect documents in a list and perform a bulk insert. This reduces the number of database round trips, which can greatly enhance performance.

Final Results (Post Optimization)

- Original Time for Fetching Students in Course: 0.0027 seconds
- Optimized Time for Fetching Students in Course: 0.0023 seconds
- Original Time for Average Students per Instructor: 0.0036 seconds
- Optimized Time for Average Students per Instructor: 0.0026 seconds
- Original Time for Courses by Department: 0.0022 seconds
- Optimized Time for Courses by Department: 0.0018 seconds
- Original Time for Total Students per Department: 0.0023 seconds
- Optimized Time for Total Students per Department: 0.0025 seconds
- Original Time for Instructors Teaching BTech CSE Core Courses: 0.0023 seconds
- Optimized Time for Instructors Teaching BTech CSE Core Courses: 0.0025 seconds
- Original Time for Top-10 Courses by Enrollment: 0.0021 seconds
- Optimized Time for Top-10 Courses by Enrollment: 0.0025 seconds

Thank You!