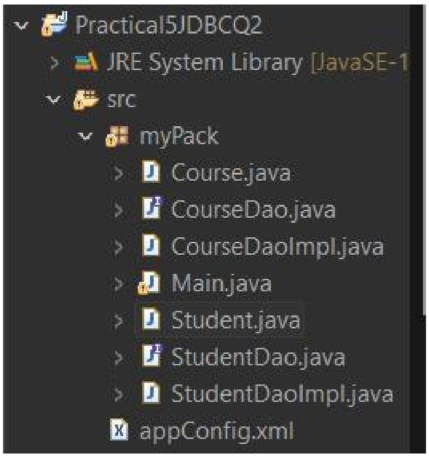
Practical No. 05
JDBC Data Access with Spring using Oracle/MySQL database

Q.1.Write a Java application for inserting and selecting multiple records from database where Student table consist of information like studId, studNm, courseId and Course table consist of information like courseId, courseNm, duration, startdt.



```
Practical5JDBCQ2
  JRE System Library [JavaSE-1
  src
    myPack
      Course.java
      CourseDao.java
      CourseDaoImpl.java
      Main.java
      Student.java
      StudentDao.java
      StudentDaoImpl.java
    appConfig.xml
```

Code:

Course.java

```
package myPack;

import java.util.Date;

public class Course {
   private int courseId;
   private String courseName;
   private String duration;
   private Date startDate;

   public Course() {
                  super();
         }
         public Course(int courseId, String courseName, String duration, Date
startDate) {
                  super();
                  this.courseId = courseId;
                  this.courseName = courseName;
                  this.duration = duration;
                  this.startDate = startDate;
         }
         @Override
         public String toString() {
                  return "Course [courseId=" + courseId + ", courseName=" +
courseName + ", duration=" + duration + ", startDate="
                                    + startDate + "]";
         }
         public int getCourseId() {
                  return courseId;
         }
         public void setCourseId(int courseId) {
                  this.courseId = courseId;
         }
```

```
        public String getCourseName() {
                return courseName;
        }
        public void setCourseName(String courseName) {
                this.courseName = courseName;
        }
        public String getDuration() {
                return duration;
        }
        public void setDuration(String duration) {
                this.duration = duration;
        }
        public Date getStartDate() {
                return startDate;
        }
        public void setStartDate(Date startDate) {
                this.startDate = startDate;
        }
}
```

CourseDao.java

```
package myPack;

import java.util.List;

public interface CourseDao {
    void create(Course course);
    Course findById(int courseId);
    List<Course> findAll();
}
```

CourseDaoImpl.java

```
package myPack;

import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class CourseDaoImpl implements CourseDao {

    private final JdbcTemplate jdbcTemplate;

    public CourseDaoImpl(JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }

    @Override
    public void create(Course course) {
        String sql = "INSERT INTO course (courseId, courseNm, duration, startdt)
VALUES (?, ?, ?, ?)";
        jdbcTemplate.update(sql, course.getCourseId(), course.getCourseName(),
course.getDuration(), course.getStartDate());
    }
```

```
   @SuppressWarnings("deprecation")
         @Override
   public Course findById(int courseId) {
      String sql = "SELECT * FROM course WHERE courseId = ?";
      return jdbcTemplate.queryForObject(sql, new Object[]{courseId}, (rs, rowNum)
->
            new Course(rs.getInt("courseId"), rs.getString("courseNm"),
rs.getString("duration"), rs.getDate("startdt")));
   }

   @Override
   public List<Course> findAll() {
      String sql = "SELECT * FROM course";
      return jdbcTemplate.query(sql, (rs, rowNum) ->
            new Course(rs.getInt("courseId"), rs.getString("courseNm"),
rs.getString("duration"), rs.getDate("startdt")));
   }
}
```

Student.java

```
package myPack;

public class Student {
   private int studId;
   private String studName;
   private int courseId;

         public int getStudId() {
                  return studId;
         }
         public void setStudId(int studId) {
                  this.studId = studId;
         }
         public String getStudName() {
                  return studName;
         }
         public void setStudName(String studName) {
                  this.studName = studName;
         }
         public int getCourseId() {
                  return courseId;
         }
         public void setCourseId(int courseId) {
                  this.courseId = courseId;
         }
         public Student(int studId, String studName, int courseId) {
                  super();
                  this.studId = studId;
                  this.studName = studName;
                  this.courseId = courseId;
         }
         @Override
         public String toString() {
                  return "Student [studId=" + studId + ", studName=" + studName + ",
courseId=" + courseId + "]";
         }
         public Student() {
                super();
        }
}
```

StudentDao.java

```
package myPack;

import java.util.List;

public interface StudentDao {
    void create(Student student);
    Student findById(int studId);
    List<Student> findAll();
    void findAllWithCourse();
}
```

StudentDaoImpl.java

```
package myPack;

import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.support.rowset.SqlRowSet;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public class StudentDaoImpl implements StudentDao {

    private final JdbcTemplate jdbcTemplate;

    public StudentDaoImpl(JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }

    @Override
    public void create(Student student) {
        String sql = "INSERT INTO student (studId, studNm, courseId) VALUES
(?, ?, ?)";
        jdbcTemplate.update(sql, student.getStudId(), student.getStudName(),
student.getCourseId());
    }

    @SuppressWarnings("deprecation")
        @Override
    public Student findById(int studId) {
        String sql = "SELECT * FROM student WHERE studId = ?";
        return jdbcTemplate.queryForObject(sql, new Object[]{studId}, (rs, rowNum) ->
                new Student(rs.getInt("studId"), rs.getString("studNm"),
rs.getInt("courseId")));
    }

    @Override
    public List<Student> findAll() {
        String sql = "SELECT * FROM student";
        return jdbcTemplate.query(sql, (rs, rowNum) ->
                new Student(rs.getInt("studId"), rs.getString("studNm"),
rs.getInt("courseId")));
```

```
    }

    @Override
    public void findAllWithCourse() {
        String sql = "SELECT s.*, c.courseNm, c.duration, c.startdt " +
                     "FROM student s " +
                     "INNER JOIN course c ON s.courseId = c.courseId";

        SqlRowSet rowSet = jdbcTemplate.queryForRowSet(sql);

        while (rowSet.next()) {
            System.out.println("Student ID: " + rowSet.getInt("studId"));
            System.out.println("Student Name: " + rowSet.getString("studNm"));
            System.out.println("Course ID: " + rowSet.getInt("courseId"));
            System.out.println("Course Name: " + rowSet.getString("courseNm"));
            System.out.println("Course Duration: " + rowSet.getString("duration"));
            System.out.println("Course Start Date: " + rowSet.getDate("startdt"));
            System.out.println("---------------");
        }
    }
}
```

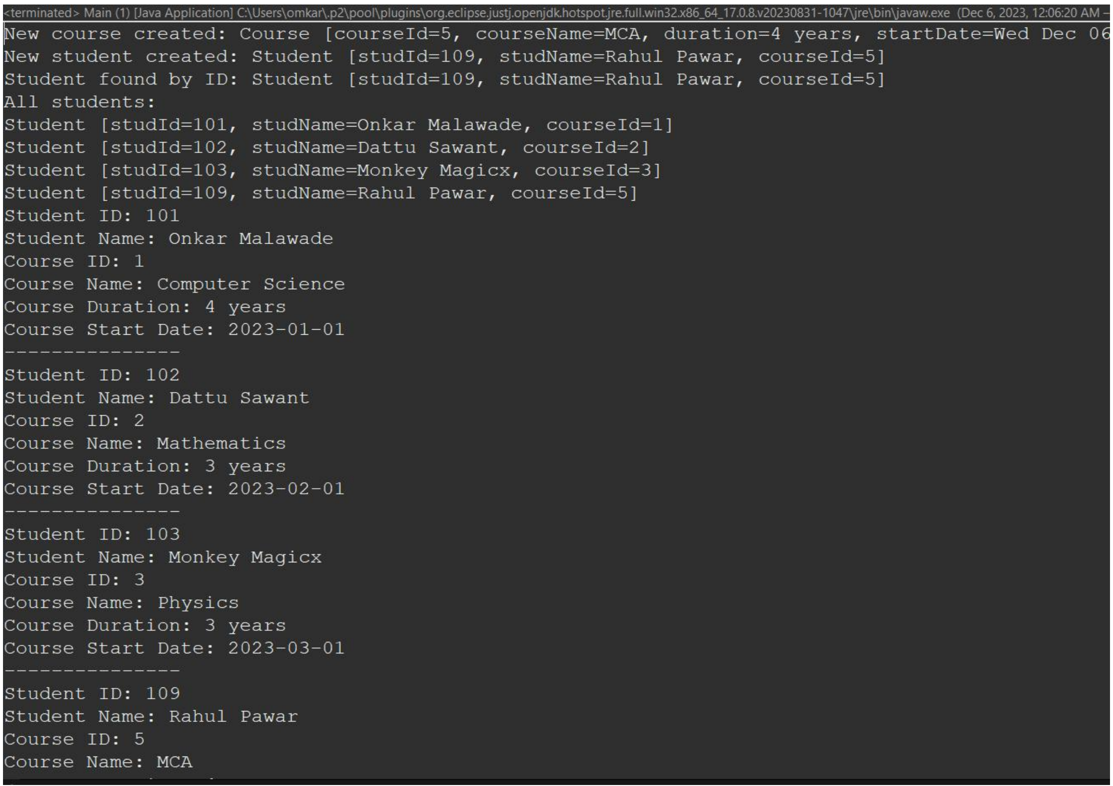
appConfig.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
       <bean id="ds" class="org.apache.commons.dbcp2.BasicDataSource">
                    <property name="driverClassName"
value="com.mysql.cj.jdbc.Driver"></property>
                    <property name="url"
value="jdbc:mysql://localhost:3306/practdb"></property>
                    <property name="username" value="root"></property>
                    <property name="password" value=""></property>
       </bean>
        <!-- JdbcTemplate Configuration -->
            <bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
            <property name="dataSource" ref="ds" />
        </bean>

            <!-- CourseDao Configuration -->
                    <bean id="courseDao" class="myPack.CourseDaoImpl">
                    <constructor-arg ref="jdbcTemplate" />
                    </bean>

                    <!-- StudentDao Configuration -->
                    <bean id="studentDao" class="myPack.StudentDaoImpl">
                    <constructor-arg ref="jdbcTemplate" />
                    </bean>
</beans>
```

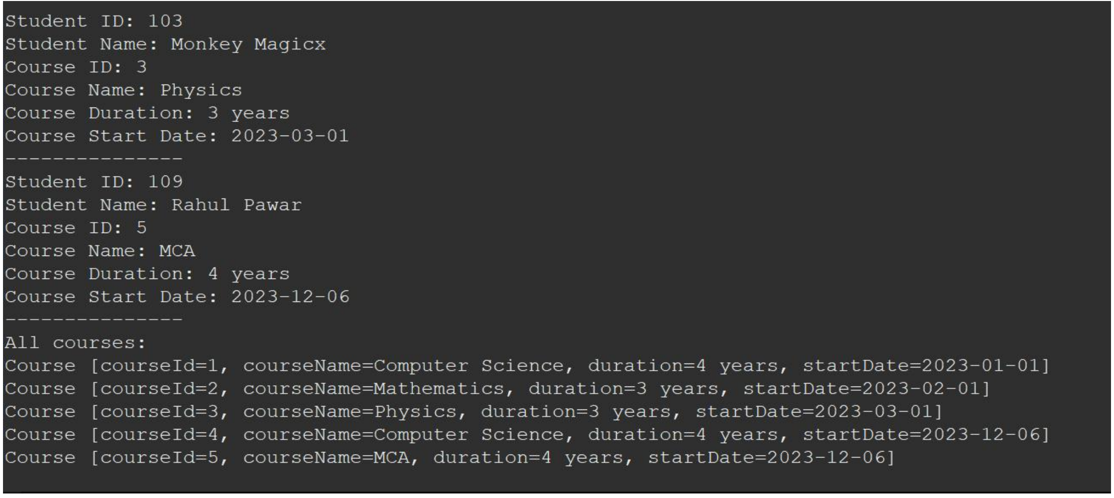
Main.java

```
package myPack;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.List;

public class Main {

    public static void main(String[] args) {
        ApplicationContext context = new
ClassPathXmlApplicationContext("appConfig.xml");
        StudentDao studentDao = context.getBean("studentDao", StudentDao.class);
        CourseDao courseDao = context.getBean("courseDao", CourseDao.class);

        // Insert a new course
        Course newCourse = new Course();
        newCourse.setCourseId(5); // Set the courseId explicitly
        newCourse.setCourseName("MCA");
        newCourse.setDuration("4 years");
        newCourse.setStartDate(new java.util.Date());
        courseDao.create(newCourse);
        System.out.println("New course created: " + newCourse);

        // Insert a new student
        Student newStudent = new Student();
        newStudent.setStudId(109); // Set the studId explicitly
        newStudent.setStudName("Rahul Pawar");
        newStudent.setCourseId(newCourse.getCourseId());
        studentDao.create(newStudent);
        System.out.println("New student created: " + newStudent);

        // Find student by ID
        int studId = newStudent.getStudId();
        Student foundStudent = studentDao.findById(studId);
        System.out.println("Student found by ID: " + foundStudent);

        // Find all students
        List<Student> allStudents = studentDao.findAll();
        System.out.println("All students:");
        for (Student student : allStudents) {
            System.out.println(student);
        }

        // Find all students with associated courses using INNER JOIN
        studentDao.findAllWithCourse();


        // Find all courses
        List<Course> allCourses = courseDao.findAll();
        System.out.println("All courses:");
        for (Course course : allCourses) {
            System.out.println(course);
        }
    }
}
```

Output:



```
<terminated> Main (1) [Java Application] C:\Users\omkar\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe  (Dec 6, 2023, 12:06:20 AM –
New course created: Course [courseId=5, courseName=MCA, duration=4 years, startDate=Wed Dec 06
New student created: Student [studId=109, studName=Rahul Pawar, courseId=5]
Student found by ID: Student [studId=109, studName=Rahul Pawar, courseId=5]
All students:
Student [studId=101, studName=Onkar Malawade, courseId=1]
Student [studId=102, studName=Dattu Sawant, courseId=2]
Student [studId=103, studName=Monkey Magicx, courseId=3]
Student [studId=109, studName=Rahul Pawar, courseId=5]
Student ID: 101
Student Name: Onkar Malawade
Course ID: 1
Course Name: Computer Science
Course Duration: 4 years
Course Start Date: 2023-01-01
---------------
Student ID: 102
Student Name: Dattu Sawant
Course ID: 2
Course Name: Mathematics
Course Duration: 3 years
Course Start Date: 2023-02-01
---------------
Student ID: 103
Student Name: Monkey Magicx
Course ID: 3
Course Name: Physics
Course Duration: 3 years
Course Start Date: 2023-03-01
---------------
Student ID: 109
Student Name: Rahul Pawar
Course ID: 5
Course Name: MCA
```



```
Student ID: 103
Student Name: Monkey Magicx
Course ID: 3
Course Name: Physics
Course Duration: 3 years
Course Start Date: 2023-03-01
---------------
Student ID: 109
Student Name: Rahul Pawar
Course ID: 5
Course Name: MCA
Course Duration: 4 years
Course Start Date: 2023-12-06
---------------
All courses:
Course [courseId=1, courseName=Computer Science, duration=4 years, startDate=2023-01-01]
Course [courseId=2, courseName=Mathematics, duration=3 years, startDate=2023-02-01]
Course [courseId=3, courseName=Physics, duration=3 years, startDate=2023-03-01]
Course [courseId=4, courseName=Computer Science, duration=4 years, startDate=2023-12-06]
Course [courseId=5, courseName=MCA, duration=4 years, startDate=2023-12-06]
```