

## 06 - RowMapper

The RowMapper is a functional interface in Spring JDBC used to map rows of the ResultSet to objects.

- It simplifies the handling of SELECT queries, allowing data from the database to be seamlessly converted into objects.
- It provides the method `mapRow()` that needs to be implemented to map each row.

### 👉 `mapRow(ResultSet rs, int rowNum):`

- This method is executed for each row in the ResultSet.
- It contains two parameters:
  - **ResultSet rs:** The result of the SQL query containing the data.
  - **int rowNum:** The current row number in the ResultSet.
- It returns a Java object representing the current row, populated with data from the ResultSet.

### 👉 `jdbcTemplate.query():`

- This method is used to perform SELECT operations in Spring JDBC.
- It requires two parameters:
  1. **SQL Query:** The SQL query that need to be executed.
  2. **RowMapper:** The object responsible for mapping each row of the ResultSet to a Java object.

## Example:

### *StudentRepo.java*

```
package com.telusko.SpringJDBCEx.repo;

@Repository
public class StudentRepo {
    private JdbcTemplate jdbc;
    @Autowired
    public void setJdbc(JdbcTemplate jdbc) {
        this.jdbc = jdbc;
    }

    public void save(Student s) {
        String sql = "INSERT INTO student(rollno, name, marks) VALUES(?, ?, ?)";
        int rows = jdbc.update(sql, s.getRollNo(), s.getName(), s.getMarks());
        System.out.println(rows + " rows affected");
    }

    // Find all students (using lambda expression)
    public List<Student> findAll() {
        String sql = "SELECT * FROM student";
        RowMapper<Student> mapper = (rs, rowNum) -> {
            Student s = new Student();

            s.setRollNo(rs.getInt("rollno"));
            s.setName(rs.getString("name"));
            s.setMarks(rs.getInt("marks"));
            return s;
        };
        return jdbc.query(sql, mapper);
    }

    // Find all students (without lambda expression)
    /*
    public List<Student> findAll() {
        String sql = "SELECT * FROM student";
        RowMapper<Student> mapper = new RowMapper<Student>() {
            @Override
            public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
                Student s = new Student();
                s.setRollNo(rs.getInt("rollno"));
                s.setName(rs.getString("name"));
                s.setMarks(rs.getInt("marks"));
                return s;
            }
        };
        return jdbc.query(sql, mapper);
    } */
}
```

## Output:

```
=====|=====|/=/ / / /
:: Spring Boot ::                (v3.2.10)

2024-10-14T12:36:51.744+05:30 INFO 18608 --- [SpringJDBCEx] [      main] c.t.S.SpringJdbcExApplication : Starting Sp
2024-10-14T12:36:51.748+05:30 INFO 18608 --- [SpringJDBCEx] [      main] c.t.S.SpringJdbcExApplication : No active p
2024-10-14T12:36:52.295+05:30 INFO 18608 --- [SpringJDBCEx] [      main] com.zaxxer.hikari.HikariDataSource : HikariPool-
2024-10-14T12:36:52.443+05:30 INFO 18608 --- [SpringJDBCEx] [      main] com.zaxxer.hikari.pool.HikariPool : HikariPool-
2024-10-14T12:36:52.444+05:30 INFO 18608 --- [SpringJDBCEx] [      main] com.zaxxer.hikari.HikariDataSource : HikariPool-
2024-10-14T12:36:52.602+05:30 INFO 18608 --- [SpringJDBCEx] [      main] c.t.S.SpringJdbcExApplication : Started Spr
1 affected
[Student [rollNo=101, name=Navin, marks=78], Student [rollNo=104, name=Kiran, marks=79], Student [rollNo=105, name=Harsh, marks=68],
2024-10-14T12:36:52.645+05:30 INFO 18608 --- [SpringJDBCEx] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-
2024-10-14T12:36:52.658+05:30 INFO 18608 --- [SpringJDBCEx] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-
```

## Code Link:

<https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/4%20Spring%20JDBC/4.6%20Rowmapper>