Using a List of Values in a JdbcTemplate IN Clause

Last updated: May 11, 2024

1. Introduction

In a SQL statement, we can use the IN operator to test whether an expression matches any value in a list. As such, we can use the IN op

In this tutorial, we'll learn how to pass a list of values into the IN clause of a Spring JDBC template (/spring-jdbc-jdbctemplate) query.

2. Passing a *List* Parameter to *IN* Clause

The IN operator allows us to specify multiple values in a WHERE clause. For example, we can use it to find all employees whose id is in

```
SELECT * FROM EMPLOYEE WHERE id IN (1, 2, 3)
```

Typically, the total number of values inside the IN clause is variable. Therefore, we need to create a placeholder that can support a dyn

2.1. With *JdbcTemplate*

With *JdbcTemplate* (https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/jdbc/core/JdbcTemp

```
List<Employee> getEmployeesFromIdList(List<Integer> ids) {
    String inSql = String.join(",", Collections.nCopies(ids.size(), "?"));

    List<Employee> employees = jdbcTemplate.query(
      String.format("SELECT * FROM EMPLOYEE WHERE id IN (%s)", inSql),
      ids.toArray(),
      (rs, rowNum) -> new Employee(rs.getInt("id"), rs.getString("first_name"),
        rs.getString("last_name")));

    return employees;
}
```

In this method, we first generate a placeholder string that contains *ids.size()* '?' characters separated with commas (/java-strings-conca

```
SELECT * FROM EMPLOYEE WHERE id IN (?,?,?)
```

In the *query* method, we pass the *ids* list as a parameter to match the placeholders inside the IN clause. This way, we can execute a dy

2.2. With *NamedParameterJdbcTemplate*

Another way to handle the dynamic list of values is to use *NamedParameterJdbcTemplate* (https://docs.spring.io/spring/docs/current

```
List<Employee> getEmployeesFromIdListNamed(List<Integer> ids) {
    SqlParameterSource parameters = new MapSqlParameterSource("ids", ids);

    List<Employee> employees = namedJdbcTemplate.query(
      "SELECT * FROM EMPLOYEE WHERE id IN (:ids)",
      parameters,
      (rs, rowNum) -> new Employee(rs.getInt("id"), rs.getString("first_name"),
        rs.getString("last_name")));

    return employees;
}
```

In this method, we first construct a *MapSqlParameterSource* (https://docs.spring.io/spring/docs/current/javadoc-api/org/springfram
Under the hood, *NamedParameterJdbcTemplate* substitutes the named parameters for the '?' placeholders, and uses *JdbcTemplate* tc

3. Handling a Large *List*

When we have a large number of values in a list, we should consider alternate ways to pass them into the *JdbcTemplate* query.
For example, the Oracle database doesn't support more than 1,000 literals in an IN clause.
One way to do this is to **create a temporary table for the list**. However, different databases can have different ways to create temporar
Let's create a temporary table for the H2 database:

```
List<Employee> getEmployeesFromLargeIdList(List<Integer> ids) {
    jdbcTemplate.execute("CREATE TEMPORARY TABLE IF NOT EXISTS employee_tmp (id INT NOT NULL)");

    List<Object[]> employeeIds = new ArrayList<>();
    for (Integer id : ids) {
        employeeIds.add(new Object[] { id });
    }
    jdbcTemplate.batchUpdate("INSERT INTO employee_tmp VALUES(?)", employeeIds);

    List<Employee> employees = jdbcTemplate.query(
      "SELECT * FROM EMPLOYEE WHERE id IN (SELECT id FROM employee_tmp)",
      (rs, rowNum) -> new Employee(rs.getInt("id"), rs.getString("first_name"),
      rs.getString("last_name")));

    jdbcTemplate.update("DELETE FROM employee_tmp");

    return employees;
}
```

Here, we first create a temporary table to hold all the values of the input list. Then we insert the input list's values into the table.
In our resulting SQL statement, **the values in the IN clause are from the temporary table**, and we avoid constructing an IN clause with
Finally, after we finish the query, we can clean up the temporary table for future use.

4. Conclusion

In this article, we demonstrated how to use *JdbcTemplate* and *NamedParameterJdbcTemplate* to pass a list of values for the IN clause
As always, the source code for the article is available over on GitHub (https://github.com/eugenp/tutorials/tree/master/persistence-