Multi Param JDBC Template

Multi Param JDBC Template

Typical Hibernate Annotation:

```
@Query("select new example.DataCollector(
    s.server, a.app,i.instance)
    from AXS x join Server s on x.server = s.server
    join Application a on x.app = a.app
    join Instance i on x.instance = i.instance
    where s.server LIKE ?1%" )
    public List<DataCollector> findAllData();
```

Multi Param JDBC Template

SQL Template:

```
select * from data where id in (?,?,?,?,?) and key in (?,?,?)
```

Multi Param JDBC Template

Request Param:

```
curl -X POST
-d "key=example+1,example+2&id=1,2,3,4,5,6"
127.0.0.1:8080/test/query

curl -X POST
-d
"key=example+1&key=example+2&id=1&id=2&id=3&id=4&id=5&id=6" 127.0.0.1:8080/test/query
```

Multi Param JDBC Template

Java Code:

```java
public List<Data> query(List<Integer> ids, List<String> keys){
    int idSize = ids.size();
    List<Object> args = new ArrayList<>();
    String questionmarks[] = new String[idSize];
    Arrays.fill(questionmarks, "?");
    args.addAll(ids);
    // flatten args into array
    String SQL = "SELECT * FROM DATA WHERE id IN (" +
        join(",", questionmarks ) + ")" + ...
    args.addAll(idSize, keys);
    ...
    return jdbcTemplate.query(SQL, args.toArray(),
                                new DataMapper());
```

Multi Param JDBC Template

Java Code:

```
Class.forName("org.postgresql.Driver");
pstmt = DriverManager.getConnection(URL, USER, PASSWORD)
    .prepareStatement(
        "select id from data where id in (%s)",
            join(",", questionmarks);

int cnt = 1; for (int id : ids)
    pstmt.setInt(cnt++, id);
rs = pstmt.executeQuery();
    wile (rs.next())
        results.add(rs.getInt("id"));
```

Multi Param JDBC Template

Runtime Error:

```
org.springframework.jdbc.BadSqlGrammarException:
org.postgresql.util.PSQLException
    ERROR: operator does not exist: integer = character varying
    Hint: No operator matches the given name and argument types.
```

Multi Param JDBC Template

Vendor-specific workaround: <u>PostgreSQL</u> (<u>YugaByte</u>):

```java
public List<BackendData> querySimilarTo(List<String> keys) {
    return jdbcTemplate.query("SELECT * FROM DATA WHERE key
            SIMILAR TO ?",
            new String[] { String.format("(%s)",
        String.join("|", keys)) },
                new BackendDataMapper());
```

Multi Param JDBC Template

Vendor-specific workaround: <u>MySQL</u>:

```java
    public List<BackendData> queryRegexpOfSetKeys(List<String>
keys) {
        return jdbcTemplate.query("SELECT * FROM DATA WHERE `key`
                    REGEXP ?",
            new String[] { String.format("(%s)",
            String.join("|", keys)) },
                new BackendDataMapper());
```