

# Database SQL Query List Action

- [Introduction](#)
- [Database SQL Query List Action Properties](#)
  - [Configure Database SQL Query List Action](#)
- [Related Database SQL Binders & Useful Links](#)
- [Download Demo App](#)



## Prevent SQL Injection

When using [Hash Variable](#) that uses URL parameter or user-inputted value in the SQL query, ensure that these hash variable(s) are **escaped** in the query!

Make use of hash variable escape keywords, see [Hash Variable - Escaping the Resultant Hash Variable](#).

### Example of VULNERABLE query:

```
SELECT * FROM app_fd_sample_table WHERE c_value = '#requestParam.id#'
```

To fix this, use **?sql** hash variable escape:

```
SELECT * FROM app_fd_sample_table WHERE c_value = '#requestParam.id?sql#'
```

## Introduction

**Database SQL Query List Action** allows you to perform SQL queries on one (a row action) or more records (a bulk action) in your datalist. You can specify which database to perform the SQL function, either the current Joget database (default datasource) or a custom datasource (external database).

Database SQL Query List Action can be used to delete records or perform an update on one or more records based on user selection in the datalist checkboxes.

## Database SQL Query List Action Properties

### Configure Database SQL Query List Action

List Builder

Test v1: List - s (Published)

Data

Design

Properties

Preview

Columns / Filters

Created By

Created By Name

Date Created

Date Modified

ID

Modified By

Modified By Name

Name

TextField

Actions

BeanShell

Hyperlink

Download Report

Delete

Form Update and Process Tool Datalist Action

Database SQL Query

Name	TextField	Date Created	Date Modified
Name	TextField	21-06-2022 09:47 AM	21-06-2022 09:47 AM
Name	TextField	21-06-2022 09:47 AM	21-06-2022 09:47 AM
Name	TextField	21-06-2022 09:47 AM	21-06-2022 09:47 AM
Name	TextField	21-06-2022 09:47 AM	21-06-2022 09:47 AM
Name	TextField	21-06-2022 09:47 AM	21-06-2022 09:47 AM

Database SQL Query

Database SQL Query

Figure 1 : Database SQL Query Action Menu

Configure Database SQL Query ?

Label \*

▼

Delete (JDBC)

Confirmation Message

Are you sure?

Datasource

Default Datasource

Query \* ?

1


DELETE FROM app\_fd\_myTable

2

WHERE id = {id}

Figure 2 : Database SQL Query List Action Properties

Name	Description
Label	Datalist button label.
Confirmation Message	Confirmation message before performing action, for example "Are you sure?".
Datasource	<div>Target database to execute SQL statements on.</div> <div>Choices:-<ul style="list-style-type: none"><li>● Custom Datasource<ul style="list-style-type: none"><li>○ JDBC Connection Parameters are needed for this choice.</li></ul></li><li>● Default Datasource<ul style="list-style-type: none"><li>○ Points to the current database your copy of Joget currently connects to.</li></ul></li></ul></div>
Custom JDBC Driver	<div>JDBC driver name.</div> <div>Example values:<ul style="list-style-type: none"><li>● com.mysql.jdbc.Driver (MySQL)</li><li>● oracle.jdbc.driver.OracleDriver (Oracle)</li><li>● com.microsoft.sqlserver.jdbc.SQLServerDriver (Microsoft SQL Server)</li></ul></div> <div>Only applicable to "Custom Datasource" option.</div>
Custom JDBC URL	<div>Database connection URL.</div> <div>Example: jdbc:mysql://localhost:3306/jwdb</div> <div>Only applicable to "Custom Datasource" option.</div>

Custom JDBC Username	<p>Database username.</p> <p>Example: root</p> <p>Only applicable to "Custom Datasource" option.</p>
Custom JDBC Password	<p>Specified database user's password.</p> <p>Only applicable to "Custom Datasource" option.</p> <div> <b>Test the connection parameters</b><p>Click on the "Test Connection" button at the bottom of the page to quickly test out your configurations.</p></div>



If a column name contains reserved keywords, do ensure it is encapsulated properly.

For example for MySQL, if the column identifier itself contains a dot symbol ( . ), it should be encapsulated like this:

```
SELECT `myAppName.myColumn` FROM app_fd_myTable;
```

Insert your SQL statement here. Use syntax like {id} in query to inject the selected row key. Use {uuid} to generate a unique id (or primary key). Examples:

#### Example

```
INSERT INTO
  app_fd_sample (id, c_clicked)
VALUES
  (
    {uuid}, {id}
  )
```

#### Example

```
UPDATE
  app_fd_sample
SET
  c_clicked = CONCAT(c_clicked, ',', {id})
WHERE
  id = {id}
```

#### Example

```
DELETE
FROM
  app_fd_myTable
WHERE
  id = {id}
```



#### Table & Column Naming

- For database tables created by Joget Forms, Joget adds a "c\_" in front of table column names (or "t\_" if your column name starts with a number) and "app\_fd\_" in front of database table names.
- If you use environment hash variables to store SQL query strings, in your hash variable, use "?noescape" to escape SQL query strings in JDBC binders to prevent the "<>" "not equal" operator from being converted, i.e. disables XSS prevention checking. [Read here for more information..](#)



#### How it works?

The special parameters {id} and {uuid} will be replaced with actual values through the use of [PreparedStatement](#). As you can see from the example above, there is no need to encapsulate both of these special keywords with quotes.

## Related Database SQL Binders & Useful Links

- [Database SQL Query Options](#)
- [Database SQL Query](#)
- [Database SQL Query List Data Store](#)
- [Understanding JDBC Errors](#)

# Download Demo App

[APP\\_datalist\\_using\\_jdbc\\_dx\\_kb.jwa](#)