Clean Up Unused Columns in Form Data Table

- Problem Statement
- Understanding the Table Structure
- Understanding the Data
- Finding Usages of the Table
- Cleaning Up the Table through the Forms
- Cleaning Up the Database Table
- Generating Hibernate Mapping File

Problem Statement

During development, we may be creating and removing fields from our form design. This would eventually lead to having unused columns in the database table. In this exercise, we will discuss how to organize a table structure and eventually remove the unused columns.

For the purpose of this exercise, we will be focusing on the table named **sample**. In Joget's database, the table name is **app_fd_sample**.

Understanding the Table Structure

Issue the following SQL to find out the exact structure of the table. With this information on hand, we would be able to better understand what columns are needed to be retained.

Table Structure

```
mysql> describe app_fd_sample;
+------------------------+--------------+------+-----+---------+-------+
| Field                  | Type         | Null | Key | Default | Extra |
+------------------------+--------------+------+-----+---------+-------+
| id                     | varchar(255) | NO   | PRI | NULL    |       |
| dateCreated            | datetime     | YES  |     | NULL    |       |
| dateModified           | datetime     | YES  |     | NULL    |       |
| c_attachment           | longtext     | YES  |     | NULL    |       |
| c_title                | longtext     | YES  |     | NULL    |       |
| createdBy              | varchar(255) | YES  |     | NULL    |       |
| createdByName          | varchar(255) | YES  |     | NULL    |       |
| modifiedBy             | varchar(255) | YES  |     | NULL    |       |
| modifiedByName         | varchar(255) | YES  |     | NULL    |       |
| c_description          | longtext     | YES  |     | NULL    |       |
| c_approver_signature   | longtext     | YES  |     | NULL    |       |
| c_approver_2_signature | longtext     | YES  |     | NULL    |       |
| c_status               | longtext     | YES  |     | NULL    |       |
| c_field1               | longtext     | YES  |     | NULL    |       |
| c_address              | longtext     | YES  |     | NULL    |       |
| c_contact              | longtext     | YES  |     | NULL    |       |
| c_name                 | longtext     | YES  |     | NULL    |       |
| c_approval_status      | longtext     | YES  |     | NULL    |       |
| c_last_approve         | longtext     | YES  |     | NULL    |       |
| c_field2               | longtext     | YES  |     | NULL    |       |
| c_desc                 | longtext     | YES  |     | NULL    |       |
| c_field3               | longtext     | YES  |     | NULL    |       |
| c_city                 | longtext     | YES  |     | NULL    |       |
| c_state                | longtext     | YES  |     | NULL    |       |
| c_approverUsername     | longtext     | YES  |     | NULL    |       |
| c_identification       | longtext     | YES  |     | NULL    |       |
| c_query                | longtext     | YES  |     | NULL    |       |
| c_result               | longtext     | YES  |     | NULL    |       |
| c_country_list         | longtext     | YES  |     | NULL    |       |
+------------------------+--------------+------+-----+---------+-------+
29 rows in set (0.00 sec)
```

Some obvious column that we should purge would be:-

- c_field1
- c_field2
- c_field3

Understanding the Data

If you are doing this in a production server, then this step is absolutely needed. We should understand the data first before we perform any sort of cleanup/housekeeping. Below is a simple query to return 1 sample row. In a production environment, we should look at the entire dataset.

Understanding the Data

```
mysql> select * from app_fd_sample limit 1 \G;
*************************** 1. row ***************************
                     id: 478407c6-45cc-433a-8684-6f052e9fb5aa
           dateCreated: 2020-04-27 09:20:12
          dateModified: 2020-04-27 09:20:12
          c_attachment: NULL
               c_title: NULL
             createdBy: admin
         createdByName: Hugo Lim
            modifiedBy: admin
        modifiedByName: Hugo Lim
         c_description: NULL
  c_approver_signature: NULL
c_approver_2_signature: NULL
              c_status: NULL
              c_field1: NULL
             c_address: NULL
             c_contact: NULL
                c_name: NULL
     c_approval_status: NULL
        c_last_approve: NULL
              c_field2: NULL
                c_desc: NULL
              c_field3: NULL
                c_city: Austin
               c_state: Alabama
    c_approverUsername: NULL
      c_identification: NULL
               c_query: NULL
              c_result: NULL
        c_country_list: NULL
1 row in set (0.00 sec)
```

Finding Usages of the Table

By using the query below, we would be able to identify which app is using this table name.

Checking the usage of the table

```
mysql> select appId, appVersion, formId, name, tableName from app_form where tableName = 'sample';
+------------------+------------+----------------------+----------------------+-----------+
| appId            | appVersion | formId               | name                 | tableName |
+------------------+------------+----------------------+----------------------+-----------+
| bulkUpdateList   |          1 | sample               | Sample               | sample    |
| bulkUpdateList   |          1 | sample_approval      | Sample Approval      | sample    |
| bulkUpdateList   |          1 | sample_clarification | Sample Clarification | sample    |
| sampleV5         |          1 | gridTest             | Grid Test            | sample    |
| sampleV5         |          1 | sample               | sample               | sample    |
| validatorOneYear |          1 | sample               | Sample               | sample    |
| vcApp            |          1 | sample               | Sample               | sample    |
| vcApp            |          2 | sample               | Sample               | sample    |
| versionSample    |          1 | sample               | Sample               | sample    |
| versionSample    |          1 | sample2              | Sample2              | sample    |
+------------------+------------+----------------------+----------------------+-----------+
10 rows in set (0.00 sec)
```

As we can see from the result, there are 10 forms pointing to the table now.

Cleaning Up the Table through the Forms

With this information on hand, we can now manually go to each of the 10 forms and identify which column/field that is still needed and **delete the unused field in the form itself.**

We can swiftly open up the form in the form builder by putting replacing the parameters as shown in the screenshot below.

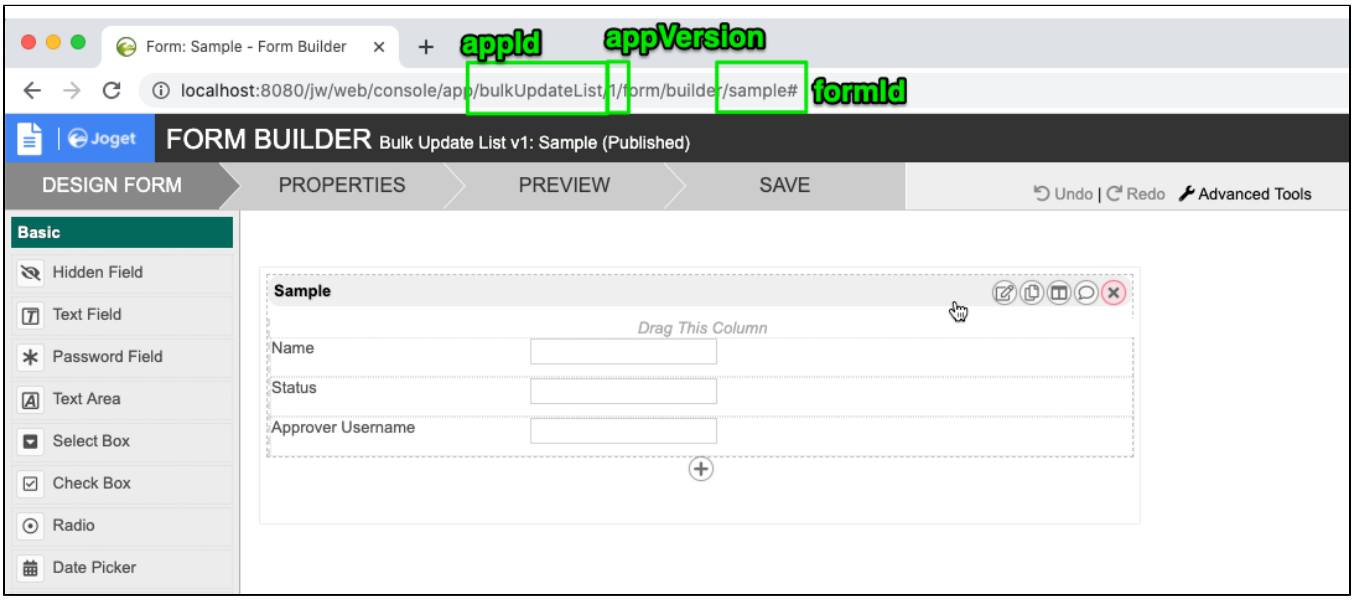
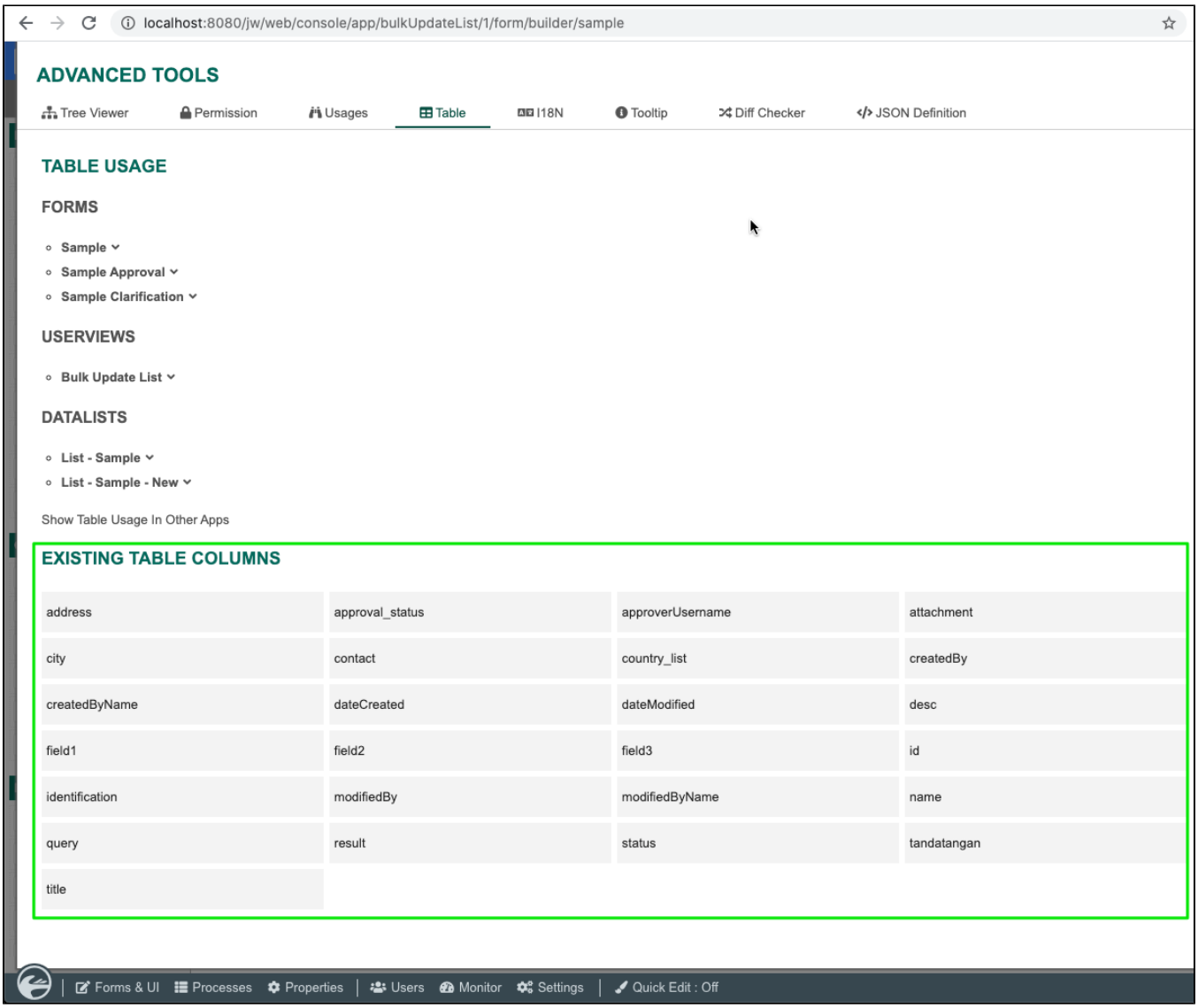


Figure 1: Inspecting the form in form builder.

If you do not have access to the database to execute the SQLs in this article, Joget has a built-in feature to find out the usages of the table too. You may click on "Advanced Tools" > "Table" to find out the usages of the same table name too.



We can only click on "Show Table Usage In Other Apps" too to find out the table's usages of the table in other apps.

Cleaning Up the Database Table

After we have gone through all the forms in the previous step, we can now move on to the database itself to remove the unused columns. This step must be done manually outside of Joget.

For example, we have determined that we would be removing the 3 fields discussed earlier. We can issue the following SQLs.

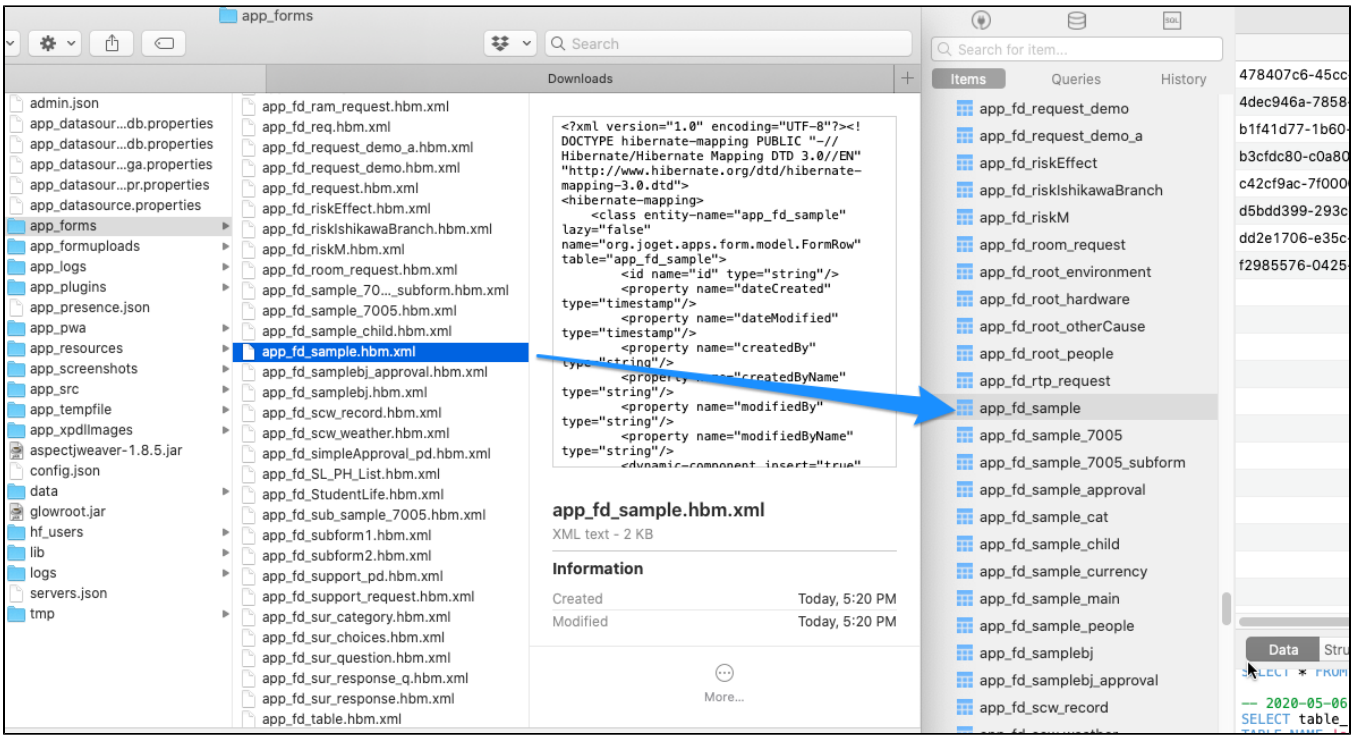
Removing columns from the database

```
ALTER TABLE `jwdb`.`app_fd_sample` DROP COLUMN `c_field1`;
ALTER TABLE `jwdb`.`app_fd_sample` DROP COLUMN `c_field2`;
ALTER TABLE `jwdb`.`app_fd_sample` DROP COLUMN `c_field3`;
```

Generating Hibernate Mapping File

We will need to delete the hibernate mapping file created by Joget in the "wflow/app_forms" folder.

Look for the file with the exact same table name. In our case, we are looking for "app_fd_sample.hbm.xml"



This step is critical. This is so that Joget would transverse through all the forms that point to the table name to generate a new mapping file, which in turn, generates any missing columns back in the database table. By not deleting this file and force a refresh, Joget would continue to assume that the table structure is still the same as always.

There is no need to restart the Joget server for this. However, we would still recommend you to restart the server as the hibernate mapping files are cached for performance reasons. If you have dropped columns that are still being referenced to in any of the forms, Joget would fail to fetch the already deleted column and this would result in an error such as the following.

Error

```
ERROR 06 May 2020 17:34:38 org.hibernate.engine.jdbc.spi.SqlExceptionHelper  - (conn=20963) Unknown column
'app_fd_sam0_.c_name' in 'field list'
ERROR 06 May 2020 17:34:38 org.joget.apps.datalist.model.DataList  - Error retrieving binder rows
org.hibernate.exception.SQLGrammarException: could not extract ResultSet
```