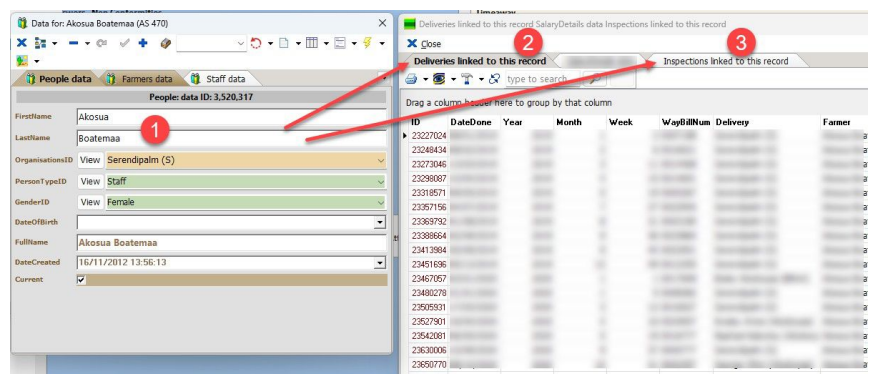


# Reallocation of Child Data: Moving child records to link to a new master record

An Orixia App contains links between "master" and "child" records. For example a Staff record might be linked to multiple Projects, or a Farmer record might be linked to multiple Deliveries and Inspections. In some situations it may be useful to reallocate child records. For example a staff member might leave, and their projects might need to be managed by a new person, or a new Farmer might take over the work of an old Farmer, and the Deliveries and Inspections might need to be "switched" to the new Farmer. Manually changing the link between multiple records is time-consuming and slow. Your administrator can run short SQL Scripts to automate this process. If this is something that needs to happen regularly, a Procedure can be created and added to your App to fully automate the process. The steps required for these processes are explained in this article.

## Reallocating data explained



Master - Child Data example: Farmer, Deliveries, Inspections

1. An example Farmer "Akosua Boatemaa" has been located.
2. Large numbers of Deliveries records are linked to this farmer, showing produce the farmer has delivered.
3. The Farmer also has multiple "Inspections" records linked to them, detailing times that staff have visited the farmer to check their farming operations.

The aim of the process is simple: To re-link all the Child "Deliveries" and "Inspections" records to a new Farmer record, all that is needed is to swap out the "Old-ID" and replace it with a new one.

## Technical details of linkages

In Orixia all data is linked on simple "integer" IDs. In the Database different data-tables use foreign key fields to allow data from one table to be linked to sets of data in other tables.

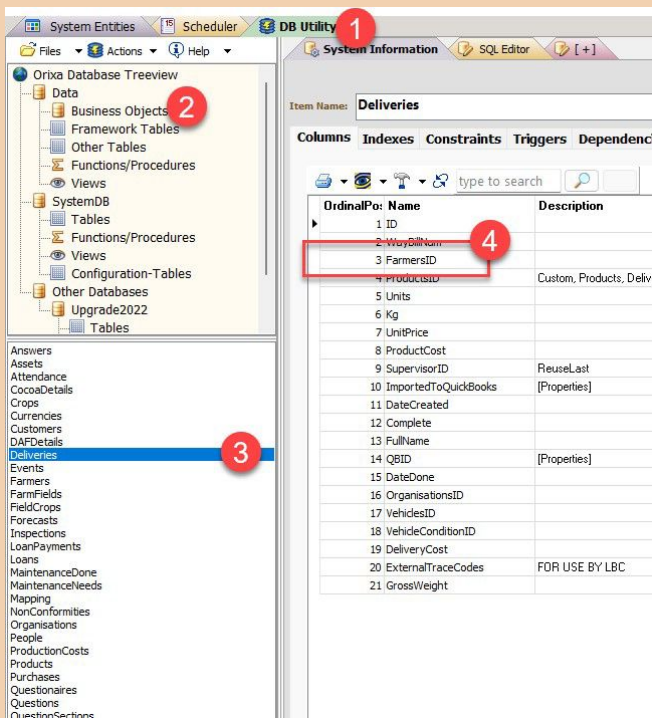
Orixia implements a very simple standard for this process. If a "Deliveries" table lists deliveries from Farmers, then the Deliveries table should include a "FarmersID" field, which will link to the ID of the farmer. In a few cases the "Link-ID field-name" will differ, but usually the "master" table-name plus "ID" is used.

### How to determine the names of the foreign-key fields

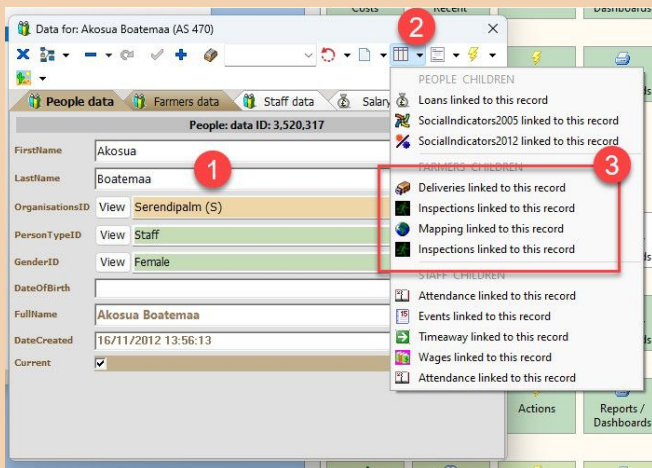
To reallocate data from a child-table to a new master record we will update the value of the foreign-key field.

Before doing this we must carefully determine the field we want to update.

1. Open the DB Utility.
2. Click on the Business Objects tab.
3. Find the Child data-table you want to check.
4. Look for the name of the foreign-key field.



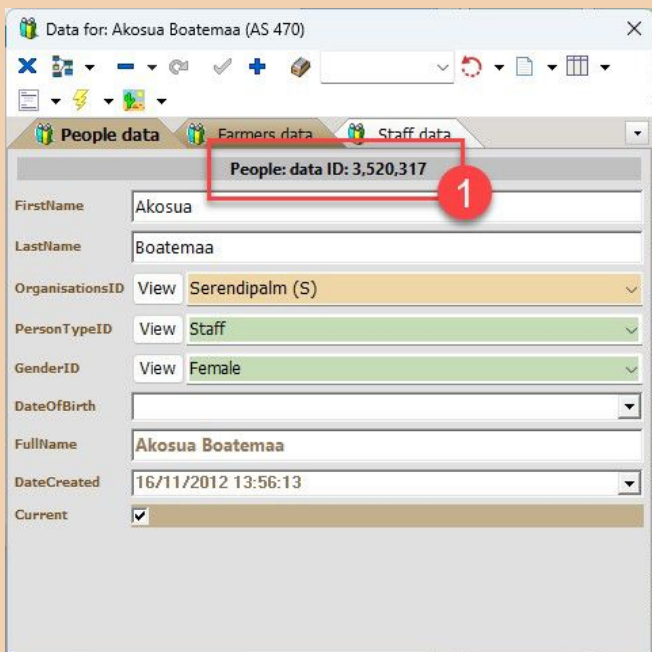
Checking the existence of the foreign key in Child data



Checking which tables contain Child data for a master record

How to determine which data-tables are children of a master record.

1. Find any master record.
2. Click on the "Show child-grids" button.
3. Child data-tables are listed. Note that they are grouped by master-table.



How to find the Master-ID you need to update.

In Orixa the master ID is displayed at the top of every edit window.

- Find the ID of the master record which is linked to the child records. This is the "Old-ID".
- Find the ID of the master record which you want to link the records to. This is the "New-ID".

## SQL Script to reallocate records

The fundamental form of the SQL is extremely simple:

```
UPDATE [Child-Table-Name]
SET [ID-Field-Name] = [New-ID-Value]
WHERE [ID-Field-Name] = [Old-ID-Value]
```

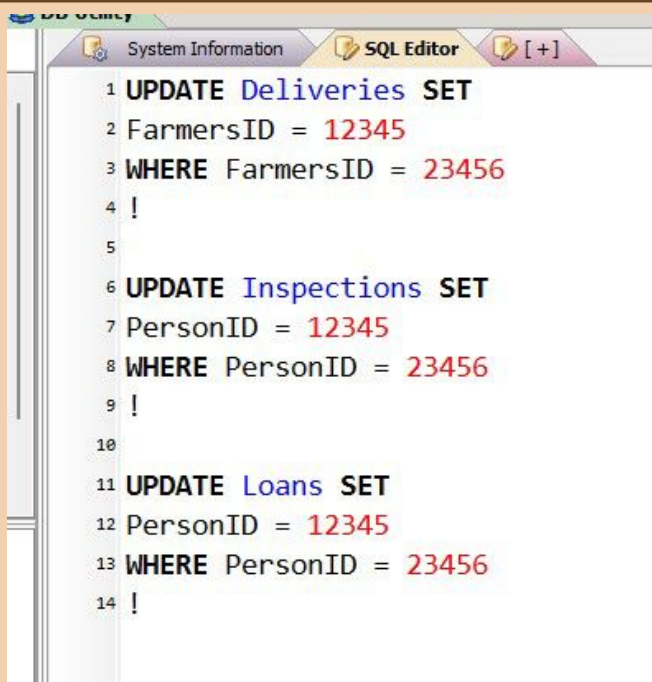
To undertake the process the admin must know:

1. The list of Child tables to be updated.
2. The list of ID-field-names
3. The Old and New IDs.

Then a SQL Script can be written to fulfil the needs of the update process.

The administrator may choose to just write this script and save it to run from time-to-time to occasionally reallocate records, or they may want to convert it into a Stored Procedure they can access, or write the Stored Procedure and surface it into the App so other users can access it.

## Example Script



```
1 UPDATE Deliveries SET
2 FarmersID = 12345
3 WHERE FarmersID = 23456
4 !
5
6 UPDATE Inspections SET
7 PersonID = 12345
8 WHERE PersonID = 23456
9 !
10
11 UPDATE Loans SET
12 PersonID = 12345
13 WHERE PersonID = 23456
14 !
```

Master - Child data example

### Steps:

1. All tables to be changed are listed.
2. Correct foreign-key field names are used.
3. Old and New ID's have been selected.

### Example SQL:

```
UPDATE Deliveries SET
FarmersID = 12345
WHERE> FarmersID = 23456
!
```

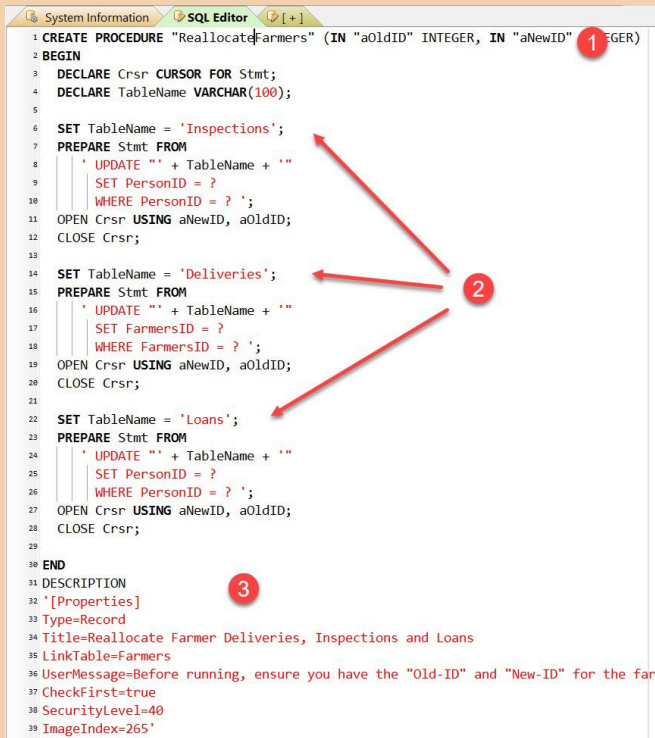
```
UPDATE Inspections SET
PersonID = 12345
WHERE PersonID = 23456
!
```

```
UPDATE Loans SET
PersonID = 12345
WHERE PersonID = 23456
!
```

## Example Stored Procedure

If a process needs to happen regularly, it should be automated with a Stored Procedure. a **DESCRIPTION** can be added to any stored procedure to surface it in the Orixia App where users can access it.

The following SQL shows a stored procedure which takes 2 parameters ("Old-ID" and "New-ID") and which reallocates records for farmer-deliveries, loans and inspections from the old to the new farmer. The description surfaces the procedure into the Orixia App. **Note that staff will still need to be trained to use the procedure, and shown how to find the "Old-ID" and "New-ID" values.** The Description can also be used to set a "SecurityLevel" so that only staff with a certain level of clearance can access the procedure.



```

1 CREATE PROCEDURE "ReallocateFarmers" (IN "aOldID" INTEGER, IN "aNewID" INTEGER)
2 BEGIN
3   DECLARE Crsr CURSOR FOR Stmt;
4   DECLARE TableName VARCHAR(100);
5
6   SET TableName = 'Inspections';
7   PREPARE Stmt FROM
8     ' UPDATE ' + TableName + '
9     SET PersonID = ?
10    WHERE PersonID = ? ';
11  OPEN Crsr USING aNewID, aOldID;
12  CLOSE Crsr;
13
14  SET TableName = 'Deliveries';
15  PREPARE Stmt FROM
16    ' UPDATE ' + TableName + '
17    SET FarmersID = ?
18    WHERE FarmersID = ? ';
19  OPEN Crsr USING aNewID, aOldID;
20  CLOSE Crsr;
21
22  SET TableName = 'Loans';
23  PREPARE Stmt FROM
24    ' UPDATE ' + TableName + '
25    SET PersonID = ?
26    WHERE PersonID = ? ';
27  OPEN Crsr USING aNewID, aOldID;
28  CLOSE Crsr;
29
30 END
31 DESCRIPTION
32 '[Properties]
33 Type=Record
34 Title=Reallocate Farmer Deliveries, Inspections and Loans
35 LinkTable=Farmers
36 UserMessage=Before running, ensure you have the "Old-ID" and "New-ID" for the far
37 CheckFirst=true
38 SecurityLevel=40
39 ImageIndex=265'

```

**SQL to create a procedure to reallocate the Deliveries, Inspections and Loans data**

### Details of the stored procedure

1. Two "IN" parameters allow a user to input the "old" and "new" ID values.
2. The script changes values in 3 data-tables. If more changes are required, just add more sections to the script.
3. The **optional** description surfaces the procedure making it visible in the App

For details of how to write **DESCRIPTIONS** to surface procedures in different parts of an Orix App read the following link:

[System Actions: Stored Procedures surfaced in the App](#)

## Example SQL

```

CREATE PROCEDURE "ReallocateFarmers" (IN "aOldID" INTEGER, IN "aNewID" INTEGER)
BEGIN
  DECLARE Crsr CURSOR FOR Stmt;
  DECLARE TableName VARCHAR(100);

  SET TableName = 'Inspections';
  PREPARE Stmt FROM
    ' UPDATE ' + TableName + '
    SET PersonID = ?
    WHERE PersonID = ? ';
  OPEN Crsr USING aNewID, aOldID;
  CLOSE Crsr;

  SET TableName = 'Deliveries';
  PREPARE Stmt FROM
    ' UPDATE ' + TableName + '
    SET FarmersID = ?
    WHERE FarmersID = ? ';
  OPEN Crsr USING aNewID, aOldID;
  CLOSE Crsr;

  SET TableName = 'Loans';
  PREPARE Stmt FROM
    ' UPDATE ' + TableName + '
    SET PersonID = ?
    WHERE PersonID = ? ';
  OPEN Crsr USING aNewID, aOldID;
  CLOSE Crsr;

  END
  DESCRIPTION
  '[Properties]
  Type=Record

```

Title=Reallocate Farmer Deliveries, Inspections and Loans

LinkTable=Farmers

UserMessage=Before running, ensure you have the "Old-ID" and "New-ID" for the farmers you want to change.

CheckFirst=true

SecurityLevel=40

ImageIndex=265'