

Synchronise a table with external data – Part 1

As with most processes in Access, there are several ways of achieving this. Whichever method is used it is recommended that a backup is made before synchronising with external data in case any problems arise.

The best solution for an individual situation will depend on various factors including:

- the number of records to be transferred or modified
- the number of records containing null values
- speed and file size

The example application attached includes all the code used in this article

It is used to synchronise a table `tblData` originally containing 200 records with another table `tblImport` containing 20,000 records. Several of the original records have missing or changed data that needs to be updated.

In addition, the code measures the time taken and the increase in file size for each method.

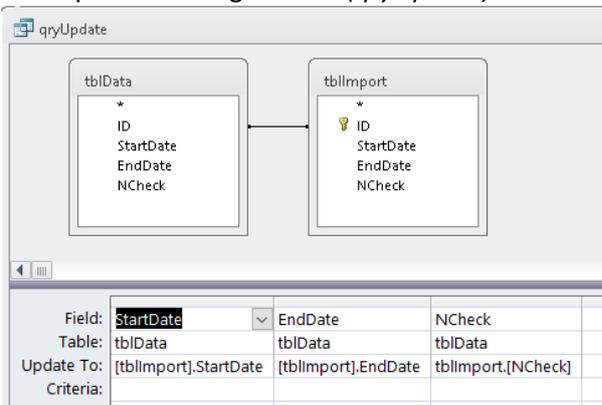
The following code assumes that:

- the records in `tblData` are being updated from `tblImport`
- both tables have the same fields and an autonumber primary key field.
- `tblImport` contains ALL records that should be imported to `tblData`



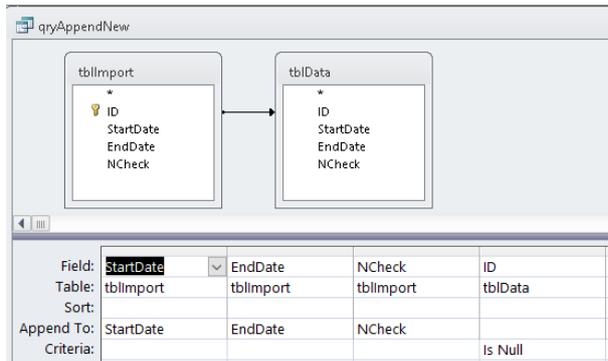
1. Update Existing / Append New / Delete Old

First update existing records (*qryUpdate*)



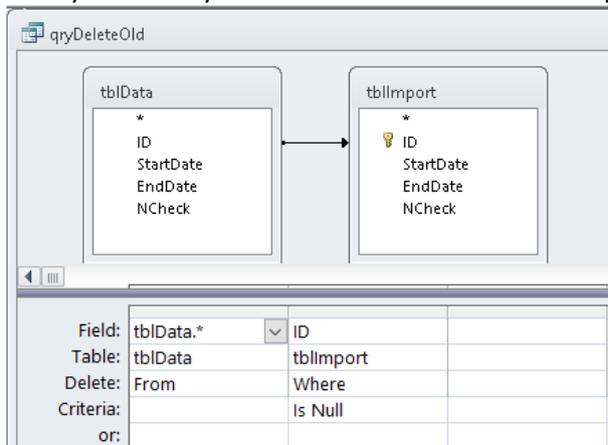
```
UPDATE tblData INNER JOIN tblImport ON tblData.ID = tblImport.ID SET tblData.StartDate = [tblImport].StartDate, tblData.EndDate = [tblImport].EndDate, tblData.NCheck = [tblImport].NCheck;
```

Next append any new records omitting the autonumber ID field (*qryAppendNew*)



```
INSERT INTO tblData ( StartDate, EndDate, NCheck )
SELECT tblImport.StartDate, tblImport.EndDate, tblImport.NCheck
FROM tblImport LEFT JOIN tblData ON tblImport.ID = tblData.ID
WHERE (((tblData.ID) Is Null));
```

Finally delete any old records that aren't in the import table (*qryDeleteOld*)



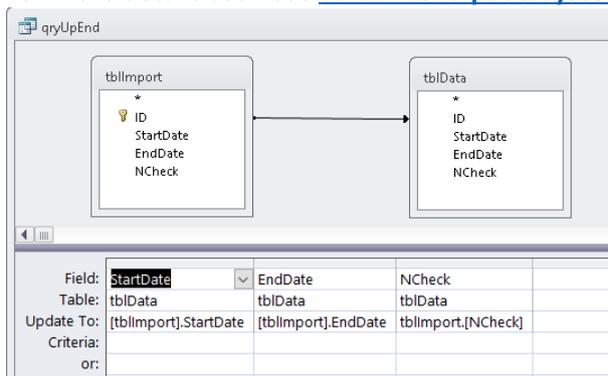
```
DELETE tblData.* FROM tblData LEFT JOIN tblImport ON tblData.ID = tblImport.ID
WHERE (((tblImport.ID) Is Null));
```

2. Combined Upend (AKA Upsert) / Delete Old

An upend or upsert query can sometimes be used to combine the append and update queries

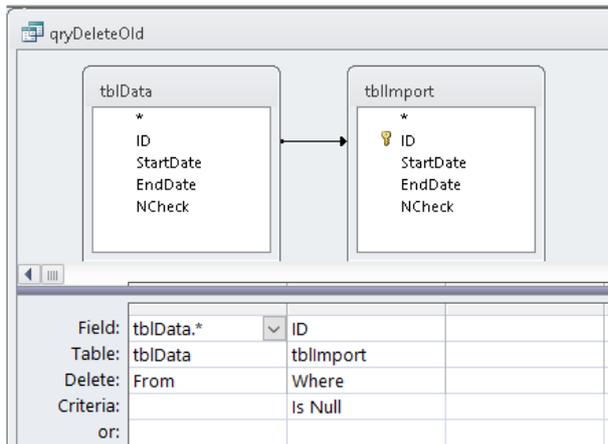
This is an append query with an outer join (*qryUpEnd*)

For more details see - see www.mendipdatasystems.co.uk/upend-query/4594428616



```
UPDATE tblData RIGHT JOIN tblImport ON tblData.ID = tblImport.ID SET tblData.StartDate =
[tblImport].StartDate, tblData.EndDate = [tblImport].EndDate, tblData.NCheck = [tblImport].NCheck;
```

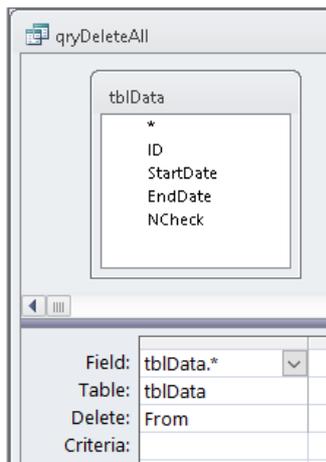
Delete any old records that aren't in the import table (*qryDeleteOld as above*)



```
DELETE tblData.* FROM tblData LEFT JOIN tblImport ON tblData.ID = tblImport.ID
WHERE (((tblImport.ID) Is Null));
```

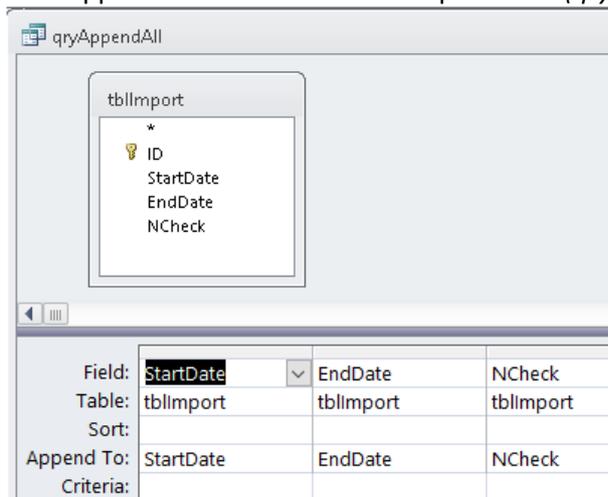
3. Delete All / Append All

First delete ALL records from tblData (*qryDeleteAll*)



```
DELETE tblData.* FROM tblData;
```

Now append all records from the import table (*qryAppendAll*)



```
INSERT INTO tblData ( StartDate, EndDate, NCheck )
SELECT tblImport.StartDate, tblImport.EndDate, tblImport.NCheck FROM tblImport;
```

NOTE:

This continues the existing ID values so if 200 records are deleted, the first appended record has ID=201

As we don't want to modify existing autonumber values, we need to reset the autonumber seed to ID=1 before appending new records.

To do so, I have used a modified version of code by Allen Browne originally at:

<http://allenbrowne.com/func-ADOX.html#ResetSeed>

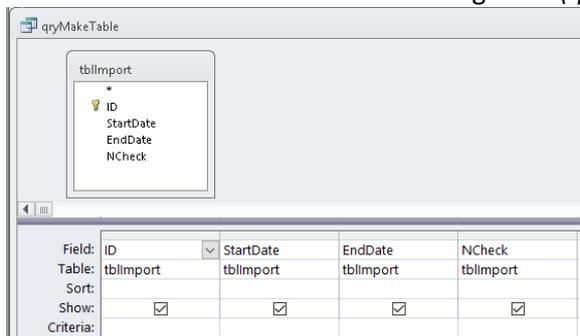
The **ResetSeed** function is in the module **modADOX** in the attached example file

NOTE: The code requires the VBA reference: **Microsoft ADO Ext 6.0 for DDL and Security**

4. Make Table / Append All

This can be done in more than one way

The first method overwrites the existing table (*qryMakeTable*)



```
SELECT tblImport.ID, tblImport.StartDate, tblImport.EndDate, tblImport.NCheck INTO tblData
FROM tblImport;
```

NOTE: The new table will not have a primary key field. This needs to be added using code or a data definition query. If you use the query designer, this can only be done in SQL view (*qryAddPrimaryKey*)

```
ALTER TABLE tblData ADD CONSTRAINT PK PRIMARY KEY (ID);
```

IMPORTANT: The following code is INCORRECT as it also resets the autonumber field (*qryAddPrimaryKeyResetAutonumber*)

```
ALTER TABLE tblData ALTER COLUMN ID COUNTER(1, 1) NOT NULL PRIMARY KEY;
```

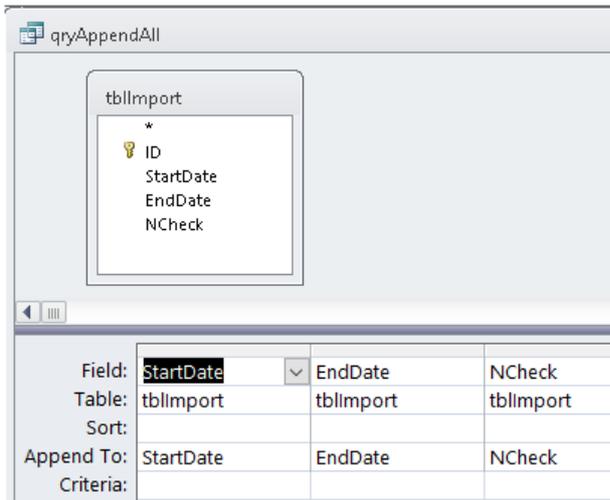
Doing this will cause the primary key to reseed starting with ID=1 with any new records

As that ID value already exists, a key violation error occurs preventing new records being appended

A MUCH BETTER approach is to use code to create new table with PK field using a data definition query (*qryCreateDataTable*)

```
CREATE TABLE tblData (ID AUTOINCREMENT NOT NULL PRIMARY KEY, StartDate DATETIME,
EndDate DATETIME, NCheck INT);
```

Next populate the table by appending all records as in method 3 (*qryAppendAll*)



```
INSERT INTO tblData ( StartDate, EndDate, NCheck )
SELECT tblImport.StartDate, tblImport.EndDate, tblImport.NCheck FROM tblImport;
```

SUMMARY

In each of the above cases, file size will increase as new records are being added
 Methods 1 and 2 will cause less file 'bloat' as the number of records being added/deleted will be smaller

Methods 3 and 4 will cause more file bloat as all records are being replaced
 The fastest method will normally be method 4 followed by method 3
 Method 2 will normally be the slowest as the upend query is less efficient for Access to process

However, unless you have a very large number of records to synchronise, the time difference may be negligible

Average Results

Workstation	Test Count	Test Type	Average Time (s)	Average File Size Change (KB)
COLIN-PC	2	1. Update Existing / Append New / Delete Old	0.66	710
COLIN-PC	2	2. Combined Upend / Delete Old	1.09	752
COLIN-PC	2	3. Delete All / Append All	0.73	770
COLIN-PC	2	4. Make Table / Append All	0.59	724

NOTE:

Indexing the fields will make the process slower as the indexes will also need to be updated.

The second part of this article looks at ways of synchronising data where there is no primary key field in the import table. This is often the case when you are importing data from an Excel or CSV file