

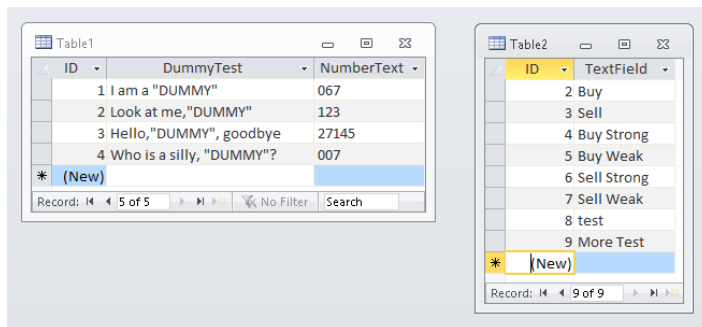
Query join types

<https://www.isladogs.co.uk/query-join-types/index.html>

When you link two or more tables together in a query, the output will depend on the type of join used between the tables. There are 5 main options to choose from

- a) Inner Join
- b) Left Outer Join
- c) Right Outer Join
- d) Union Query (AKA Full Outer Join)
- e) Cartesian Join (AKA No Join)

To illustrate the effect of each, this article uses two randomly chosen tables with 4 and 8 records respectively



ID	DummyTest	NumberText
1	I am a "DUMMY"	067
2	Look at me,"DUMMY"	123
3	Hello,"DUMMY", goodbye	27145
4	Who is a silly, "DUMMY"?	007

ID	TextField
2	Buy
3	Sell
4	Buy Strong
5	Buy Weak
6	Sell Strong
7	Sell Weak
8	test
9	More Test

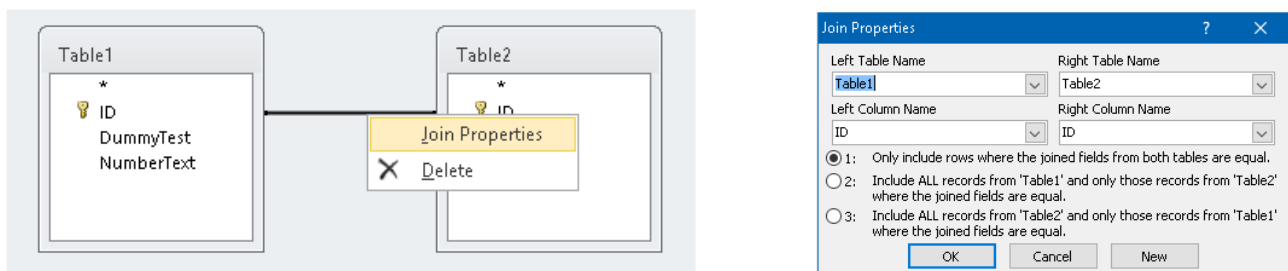
Both tables have an ID fields which will be used as the link fields in these queries

a) Inner Join

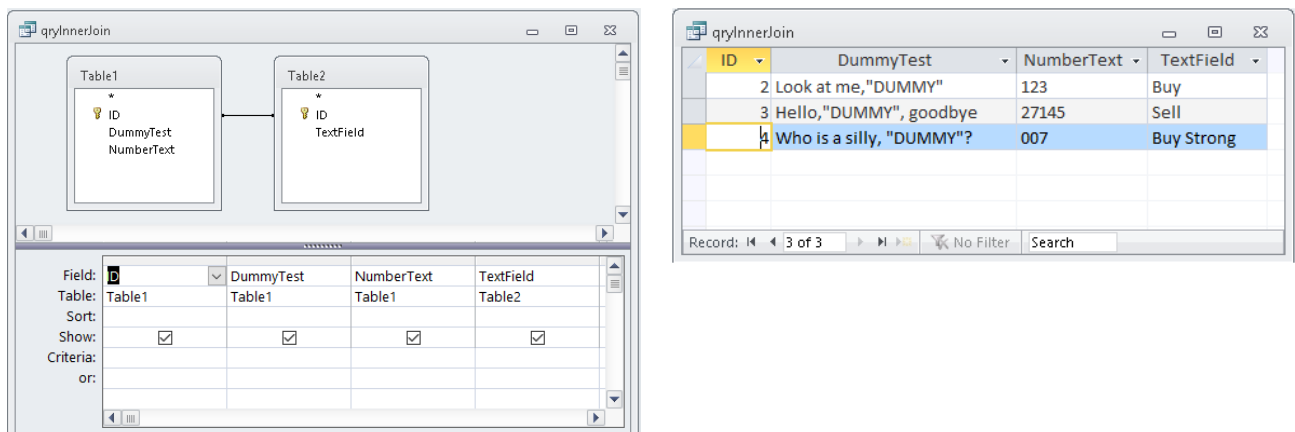
The two tables are linked by connecting the two ID fields. By default an **inner join** is created.

Right click on the joining line and select **Join Properties**.

The **Join Properties** dialog box appears and explains what the join means



When the query is run, it has 3 records as only ID 2, 3 and 4 are in both tables



ID	DummyTest	NumberText	TextField
2	Look at me,"DUMMY"	123	Buy
3	Hello,"DUMMY", goodbye	27145	Sell
4	Who is a silly, "DUMMY"?	007	Buy Strong

The query SQL is:

```
SELECT DISTINCT Table1.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField
FROM Table1 INNER JOIN Table2 ON Table1.ID = Table2.ID
```

The **Join Properties** dialog box also explains the next two options

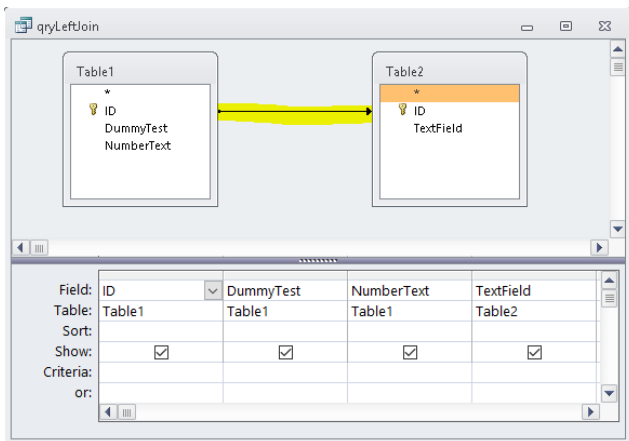
b) Left Outer Join

In this option 2 is chosen – all records from table 1 (on the left) and only matching records in table 2

Notice the direction of the arrow in the query design.

All 4 records from table 1 are listed with the corresponding data from table 2

The query design and results are:



ID	DummyTest	NumberText	TextField
1	I am a "DUMMY"	067	
2	Look at me, "DUMMY"	123	Buy
3	Hello, "DUMMY", goodbye	27145	Sell
4	Who is a silly, "DUMMY"?	007	Buy Strong
* (New)			

Record: 1 of 4

The query SQL is:

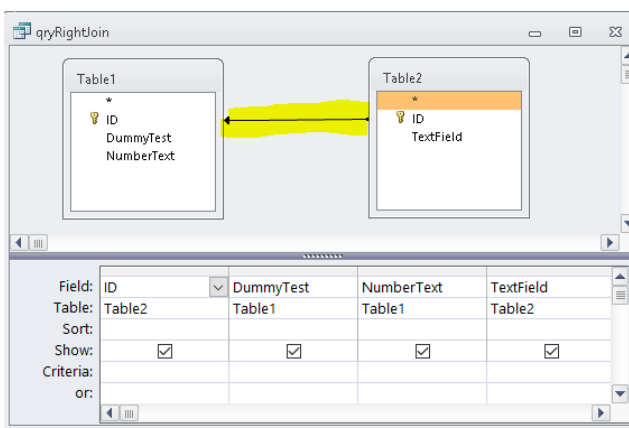
```
SELECT Table1.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField
FROM Table1 LEFT JOIN Table2 ON Table1.ID = Table2.ID;
```

c) Right Outer Join

In this option 3 is chosen – all records from table 2 (on the right) and only matching records in table 1

Notice the direction of the arrow in the query design. In this case there are 8 records in the query output

The query design and results are:



ID	DummyTest	NumberText	TextField
1	Look at me, "DUMMY"	123	Buy
2	Hello, "DUMMY", goodbye	27145	Sell
3	Who is a silly, "DUMMY"?	007	Buy Strong
4			Buy Weak
5			Sell Strong
6			Sell Weak
7			test
8			More Test
* (New)			

Record: 1 of 8

The query SQL is:

```
SELECT [Table2].ID, [Table1].DummyTest, [Table1].NumberText, [Table2].TextField
FROM Table1 RIGHT JOIN Table2 ON Table1.ID=[Table2].ID;
```

d) Union Query

Access does not support a full outer join needed to get all values from both tables
Instead a union query is used to achieve this outcome.
It is effectively two (or more) queries 'unioned' together
Union queries can only be created in SQL view

NOTE: Each part of a union query must have the same number of fields with the same datatypes.
That isn't an issue in this example.
However, you should add 'DUMMY' fields to one or both parts to achieve that where necessary.

The query SQL is:

```
SELECT Table1.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField
FROM Table1 LEFT JOIN Table2 ON Table1.ID = Table2.ID;
UNION SELECT Table2.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField
FROM Table1 RIGHT JOIN Table2 ON Table1.ID = Table2.ID;
```

The query results show all records from both tables with gaps where there is no corresponding record:

The screenshot shows the results of a union query in Access. The query is named 'qryUnion'. The results are displayed in a table with four columns: ID, DummyTest, NumberText, and TextField. The data is as follows:

ID	DummyTest	NumberText	TextField
1	I am a "DUMMY"	067	
2	Look at me,"DUMMY"	123	Buy
3	Hello,"DUMMY", goodbye	27145	Sell
4	Who is a silly, "DUMMY"?	007	Buy Strong
5			Buy Weak
6			Sell Strong
7			Sell Weak
8			test
9			More Test

e) Cartesian Join

In this case the join is deleted so we have two unconnected tables.
The query output has a separate record for every combination of records from each table
This gives a total of $4 \times 8 = 32$ records

The screenshot shows the design view of a query named 'qryCartesianJoin'. It shows two tables, Table1 and Table2, connected by a Cartesian join. The fields and their sources are as follows:

Field	Table	Sort	Show	Criteria	or
ID	Table1	Ascending	☒		
DummyTest	Table1		☒		
NumberText	Table1		☒		
ID	Table2	Ascending	☒		
TextField	Table2		☒		

The screenshot shows the results of a Cartesian join query in Access. The query is named 'qryCartesianJoin'. The results are displayed in a table with five columns: Table1.ID, DummyTest, NumberText, Table2.ID, and TextField. The data is as follows:

Table1.ID	DummyTest	NumberText	Table2.ID	TextField
1	I am a "DUMMY"	067	2	Buy
1	I am a "DUMMY"	067	3	Sell
1	I am a "DUMMY"	067	4	Buy Strong
1	I am a "DUMMY"	067	5	Buy Weak
1	I am a "DUMMY"	067	6	Sell Strong
1	I am a "DUMMY"	067	7	Sell Weak
1	I am a "DUMMY"	067	8	test
1	I am a "DUMMY"	067	9	More Test
2	Look at me,"DUMMY"	123	2	Buy
2	Look at me,"DUMMY"	123	3	Sell
2	Look at me,"DUMMY"	123	4	Buy Strong
2	Look at me,"DUMMY"	123	5	Buy Weak
2	Look at me,"DUMMY"	123	6	Sell Strong
2	Look at me,"DUMMY"	123	7	Sell Weak
2	Look at me,"DUMMY"	123	8	test
2	Look at me,"DUMMY"	123	9	More Test
3	Hello,"DUMMY", goodbye	27145	2	Buy
3	Hello,"DUMMY", goodbye	27145	3	Sell
3	Hello,"DUMMY", goodbye	27145	4	Buy Strong
3	Hello,"DUMMY", goodbye	27145	5	Buy Weak
3	Hello,"DUMMY", goodbye	27145	6	Sell Strong
3	Hello,"DUMMY", goodbye	27145	7	Sell Weak
3	Hello,"DUMMY", goodbye	27145	8	test
3	Hello,"DUMMY", goodbye	27145	9	More Test
4	Who is a silly, "DUMMY"?	007	2	Buy
4	Who is a silly, "DUMMY"?	007	3	Sell
4	Who is a silly, "DUMMY"?	007	4	Buy Strong
4	Who is a silly, "DUMMY"?	007	5	Buy Weak
4	Who is a silly, "DUMMY"?	007	6	Sell Strong
4	Who is a silly, "DUMMY"?	007	7	Sell Weak
4	Who is a silly, "DUMMY"?	007	8	test
4	Who is a silly, "DUMMY"?	007	9	More Test

The query SQL is:

```
SELECT Table1.ID, Table1.DummyTest, Table1.NumberText, Table2.ID, Table2.TextField
FROM Table1, Table2
ORDER BY Table1.ID, Table2.ID;
```

For most purposes, a Cartesian join is undesirable though it does have its uses.
For example, to populate a calendar events table with all the dates for a calendar year

Using 3 (or more) tables

Queries can also be made using one or more additional tables . For example:

The image shows three separate database tables. Table1 has columns ID, DummyTest, and NumberText. Table2 has columns ID and TextField. Table3 has columns ID and TextField2.

ID	DummyTest	NumberText
1	I am a "DUMMY"	067
2	Look at me,"DUMMY"	123
3	Hello,"DUMMY", goodbye	27145
4	Who is a silly, "DUMMY"?	007
* (New)		

ID	TextField
2	Buy
3	Sell
4	Buy Strong
5	Buy Weak
6	Sell Strong
7	Sell Weak
8	test
9	More Test
* (New)	

ID	TextField2
2	O'Neill
3	Fred's book
5	2 pieces of 8' - 2" x 4"
* (New)	

f) Inner Joins

This will only show all records that are in EACH of the tables:

The image shows a query design view for a query named qryInnerJoin3Tables. It displays three tables: Table1, Table2, and Table3. Table1 has fields ID, DummyTest, and NumberText. Table2 has fields ID and TextField. Table3 has fields ID and TextField2. The ID field is the primary key for all three tables. The query is an inner join of all three tables on the ID field.

Field:	ID	DummyTest	NumberText	TextField	TextField2
Table:	Table2	Table1	Table1	Table2	Table3
Sort:					
Show:	☒	☒	☒	☒	☒
Criteria:					
or:					

The query SQL is:

```
SELECT Table2.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField, Table3.TextField2  
FROM (Table1 INNER JOIN Table2 ON Table1.ID = Table2.ID) INNER JOIN Table3 ON Table1.ID = Table3.ID;
```

As all conditions need to be met, this often REDUCES the number of records shown

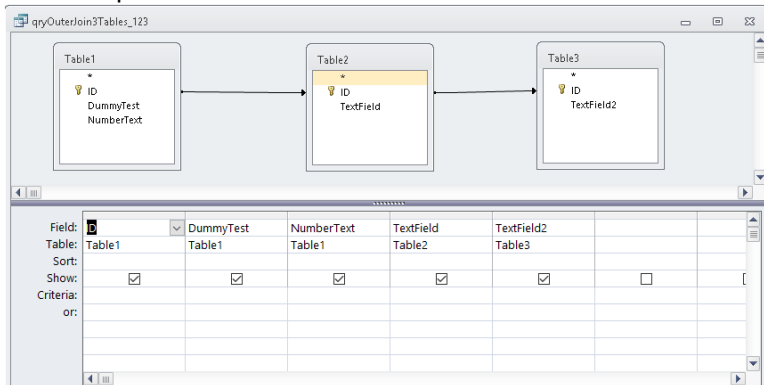
The image shows the query result view for qryInnerJoin3Tables. It displays a table with 5 columns: ID, DummyTest, NumberText, TextField, and TextField2. Only 2 records are shown, corresponding to ID 2 and ID 3, as they are the only records present in all three tables.

ID	DummyTest	NumberText	TextField	TextField2
2	Look at me,"DUMMY"	123	Buy	O'Neill
3	Hello,"DUMMY", goodbye	27145	Sell	Fred's book
* (New)				

g) Outer Joins

There are various possibilities. In each case the arrow 'flow direction' needs to be consistent

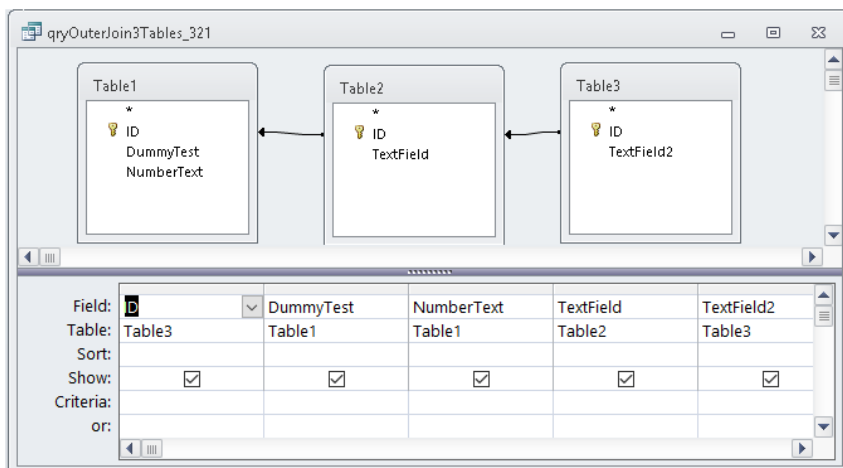
For example: Table1=> Table2=>Table3



```
SELECT Table1.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField, Table3.TextField2
FROM (Table1 LEFT JOIN Table2 ON Table1.ID = Table2.ID) LEFT JOIN Table3 ON Table2.ID = Table3.ID;
```

ID	DummyTest	NumberText	TextField	TextField2
1	I am a "DUMMY"	067		
2	Look at me,"DUMMY"	123	Buy	O'Neill
3	Hello,"DUMMY", goodbye	27145	Sell	Fred's book
4	Who is a silly, "DUMMY"?	007	Buy Strong	

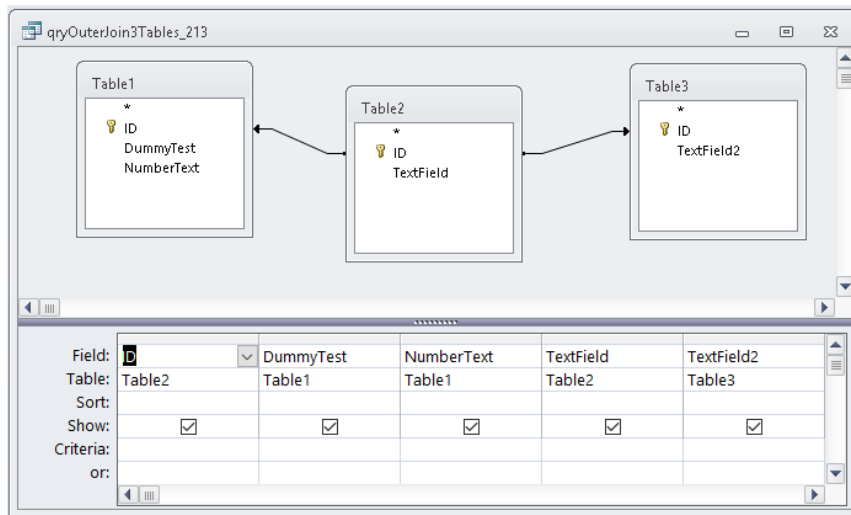
Another valid possibility is Table3=> Table2=>Table1



```
SELECT Table3.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField, Table3.TextField2
FROM Table3 LEFT JOIN (Table1 RIGHT JOIN Table2 ON Table1.ID = Table2.ID) ON Table3.ID = Table2.ID;
```

ID	DummyTest	NumberText	TextField	TextField2
2	Look at me,"DUMMY"	123	Buy	O'Neill
3	Hello,"DUMMY", goodbye	27145	Sell	Fred's book
5			Buy Weak	2 pieces of 8' - 2" x 4"

Or the flow can start from Table2 which has the most records:

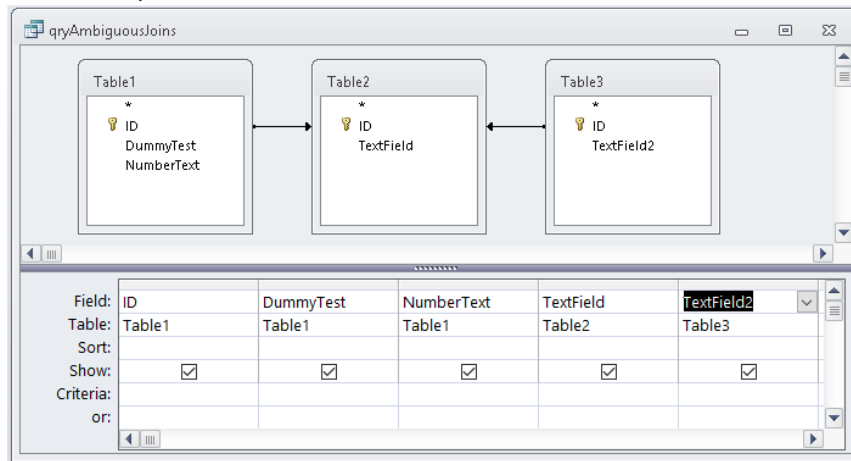


```
SELECT Table2.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField, Table3.TextField2
FROM (Table1 RIGHT JOIN Table2 ON Table1.ID = Table2.ID) LEFT JOIN Table3 ON Table2.ID = Table3.ID;
```

ID	DummyTest	NumberText	TextField	TextField2
2	Look at me, "DUMMY"	123	Buy	O'Neill
3	Hello, "DUMMY", goodbye	27145	Sell	Fred's book
4	Who is a silly, "DUMMY"?	007	Buy Strong	
5			Buy Weak	2 pieces of 8' - 2" x 4"
6			Sell Strong	
7			Sell Weak	
8			test	
9			More Test	
(New)				

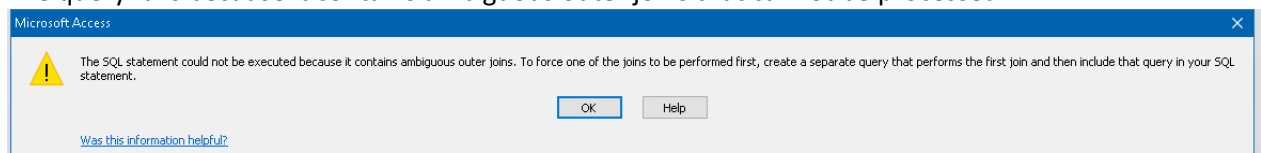
h) Ambiguous Joins

In this case, the flow direction isn't consistent:



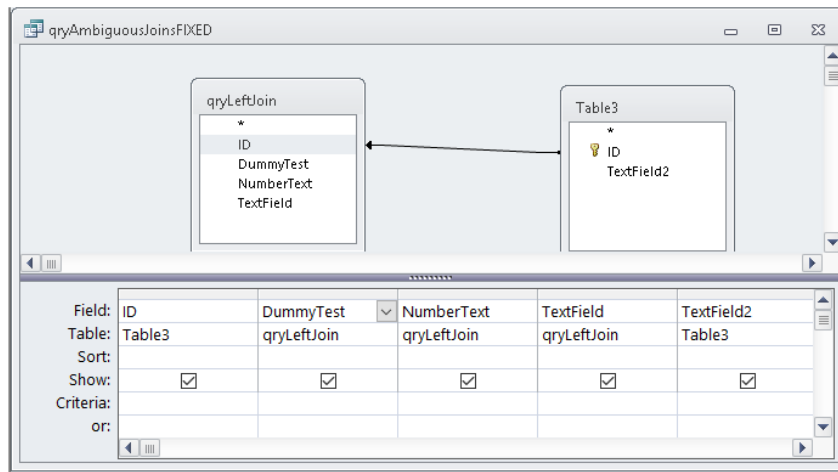
```
SELECT Table1.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField, Table3.TextField2
FROM Table3 LEFT JOIN (Table1 LEFT JOIN Table2 ON Table1.ID = Table2.ID) ON Table3.ID = Table2.ID;
```

The query fails because it contains ambiguous outer joins that cannot be processed



To fix this issue, do the query in two stages so one of the joins is processed first.

For example, use the existing qryLeftJoin and link to Table3



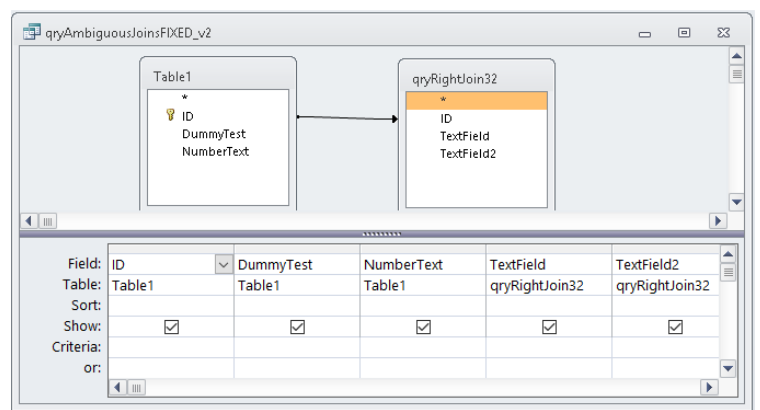
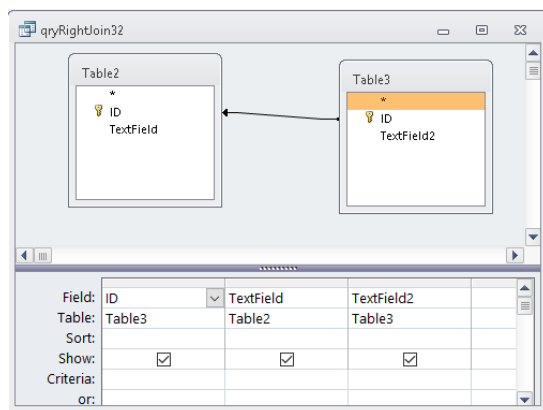
```
SELECT Table3.ID, qryLeftJoin.DummyTest, qryLeftJoin.NumberText, qryLeftJoin.TextField,
Table3.TextField2
FROM qryLeftJoin RIGHT JOIN Table3 ON qryLeftJoin.ID = Table3.ID;
```

There are 3 records

ID	DummyTest	NumberText	TextField	TextField2
2	Look at me,"DUMMY"	123	Buy	O'Neill
3	Hello,"DUMMY", goodbye	27145	Sell	Fred's book
5				2 pieces of 8' - 2" x 4"
* (New)				

Record: 1 of 3

Alternatively, create a right outer join query based on Tables 2 & 3 then link to Table1 using a left join



```
SELECT Table1.ID, Table1.DummyTest, Table1.NumberText, qryRightJoin32.TextField,
qryRightJoin32.TextField2
FROM qryRightJoin32 RIGHT JOIN Table1 ON qryRightJoin32.ID = Table1.ID;
```

This time, the result has 4 records

ID	DummyTest	NumberText	TextField	TextField2
1	I am a "DUMMY"	067		
2	Look at me,"DUMMY"	123	Buy	O'Neill
3	Hello,"DUMMY", goodbye	27145	Sell	Fred's book
4	Who is a silly, "DUMMY"?	007		
* (New)				

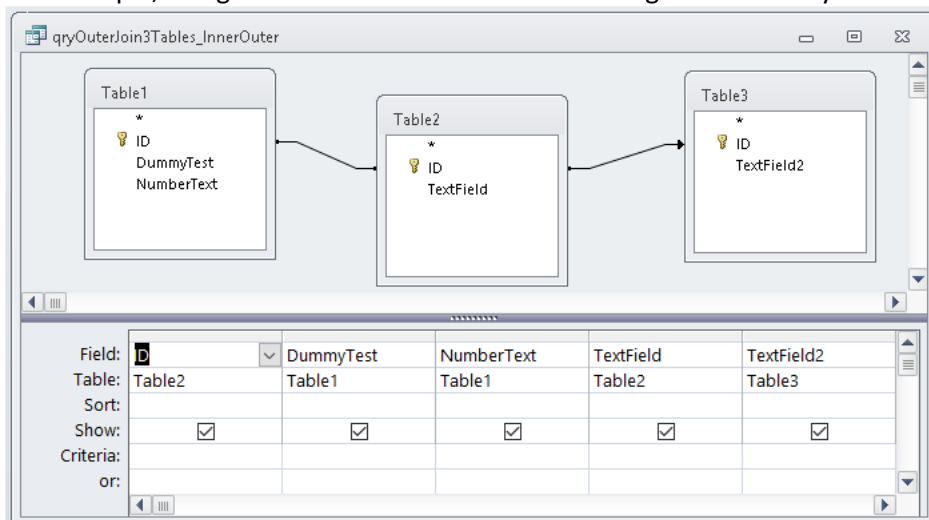
Record: 1 of 4

The fact that the outcome is different explains the original ambiguous joins query could not be processed. Access has no means of knowing how to resolve what the query means.

i) **Mixed Inner/Outer Joins**

It is also possible to use a mixture of inner and outer joins.

For example, this get all records in both table 1 & 2 together with any matching records in table 3



```
SELECT Table2.ID, Table1.DummyTest, Table1.NumberText, Table2.TextField, Table3.TextField2
FROM (Table1 INNER JOIN Table2 ON Table1.ID = Table2.ID) LEFT JOIN Table3 ON Table2.ID = Table3.ID;
```

ID	DummyTest	NumberText	TextField	TextField2
2	Look at me, "DUMMY"	123	Buy	O'Neill
3	Hello, "DUMMY", goodbye	27145	Sell	Fred's book
4	Who is a silly, "DUMMY"?	007	Buy Strong	

NOTE:

Using more tables increases the chances that the query will:

- be **READ ONLY**
- showing **duplicate records**

All the above methods can be adapted to manage cases where duplicate records are displayed.

There are 2 options :

- **Unique Values = Yes (SELECT DISTINCT)** – this automatically makes the query **READ ONLY**
- **Unique Records = Yes (SELECT DISTINCTROW)**

These will be discussed in more detail in a future article