## Extracting Data From Multiple Tables



All examples so far have retrieved data from only **one** table.

Many applications require data to be retrieved from two or more tables.

To combine columns from two or more tables into a result we must **JOIN** the tables.



JOIN combines information from two tables by forming ***pairs*** of related records from the two tables

Tables cannot be joined unless they have a **common attribute**.

The rows included in the result-set of a JOIN query are only those where ***matching columns*** in each table have equal values.



Consider the tables Staff and Branch on the handout

Branch : Table

| Bno | Street | Town | County | TelNo |
|---|---|---|---|---|
| 10 | MAIN ST | KILLARNEY | CO.KERRY | 064312556 |
| 20 | HIGH STREET | KENMARE | CO.KERRY | 064325569 |
| 30 | IVERAGH ROAD | KILLORGLIN | CO.KERRY | 0669761314 |
| 40 | THE BARRACKS | CAHIRCIVEEN | CO.KERRY | 0669488922 |
| 0 | | | | |

Staff : Table

| StaffNo | Surname | Forename | Bno | Salary |
|---|---|---|---|---|
| 1 | SMITH | JOHN | 10 | $25,000.00 |
| 2 | JONES | MARY | 30 | $35,000.00 |
| 3 | KELLY | MICHAEL | 10 | $28,500.00 |
| 4 | KENNY | JOAN | 20 | $52,000.00 |
| 5 | WOODS | JAMES | 30 | $36,000.00 |
| 6 | DUGGAN | ANNE | 40 | $85,000.00 |
| 0 | | | 0 | $0.00 |



Suppose we wish to find the staff number, name details, branch number and town where branch is located for all staff. The following result is desired:

Query1

| StaffNo | Surname | Forename | Bno | Town |
|---|---|---|---|---|
| 1 | SMITH | JOHN | 10 | KILLARNEY |
| 3 | KELLY | MICHAEL | 10 | KILLARNEY |
| 4 | KENNY | JOAN | 20 | KENMARE |
| 2 | JONES | MARY | 30 | KILLORGLIN |
| 5 | WOODS | JAMES | 30 | KILLORGLIN |
| 6 | DUGGAN | ANNE | 40 | CAHIRCIVEEN |



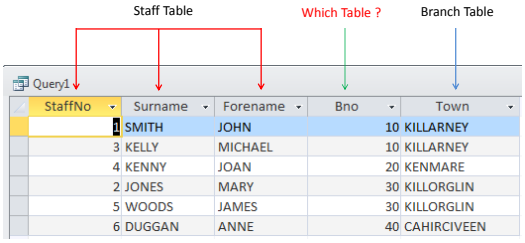
You must identify the table from which each output attribute value is to be retrieved :

StaffId, Surname and forename are attributes in the ***Staff*** table.

Town is an attribute in the ***Branch*** table.

What about the attribute Bno?
This attribute exists in both the ***Branch*** and ***Staff*** tables.

Shown on the diagram below:






The attribute Bno exists in both the ***Staff*** and the ***Branch*** tables.

Q: Which table is the value taken from?

A: Analyse the requirements carefully and then decide.

In this instance, it should be taken from the Staff table. This needs to be stated in the SELECT clause of the query.



To perform a JOIN:

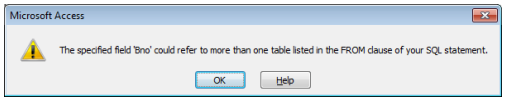
- List the required output columns in the SELECT clause.
- Include all required table names in the FROM clause (separated by commas)
- Include a WHERE clause to specify the JOIN columns (could also use JOIN – more later!)



List all staff and the town in which they work

```
SELECT StaffNo, Surname, Forename, Bno, Town
FROM Staff, Branch
WHERE Staff.Bno = Branch.Bno;
```

Problem!






Since Bno exists in both tables, we must specify in the SELECT clause which table the output value is to be taken from.

```
SELECT StaffNo, Surname, Forename, Staff.Bno, Town
FROM Staff, Branch
WHERE Staff.Bno = Branch.Bno;
```

This produces the desired output.



We can use ***table aliases*** to reduce typing!

```
SELECT StaffNo, Surname, Forename, S.Bno, Town
FROM Staff S, Branch B
WHERE S.Bno = B.Bno;
```

## Using the JOIN Operator

An alternative way of joining tables is to use the FROM clause

SELECT StaffNo, Surname, Forename, S.BNo, Town
FROM Staff S **INNER JOIN** Branch B ON S.BNo = **B**.BNo;

In MS Access, we must specify the JOIN type (in this example the JOIN type is an ***INNER*** join).

**Notice that there is no WHERE clause in this case!**



## Outer Joins

A JOIN may be:

- INNER JOIN (default)
- OUTER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN



The previous example illustrates an INNER join. INNER is the default JOIN type.

```
SELECT StaffNo, Surname, Forename, Staff.Bno, Town
FROM Staff, Branch
WHERE Staff.Bno = Branch.Bno;
```

Only rows where a matching value is found in the common attribute are included in the result set.



Suppose we wish to list all staff and the property number and town for properties which they oversee.

```
SELECT S.StaffNo,Surname, Forename,PropNo, Town
FROM Staff S, Properties P
WHERE S.StaffNo = P.StaffNo;
```



The following result is produced:

Query1

| StaffNo | Surname | Forename | PropNo | Town |
|---|---|---|---|---|
| 1 | SMITH | JOHN | 4 | KILLARNEY |
| 2 | JONES | MARY | 2 | TRALEE |
| 2 | JONES | MARY | 3 | TRALEE |
| 4 | KENNY | JOAN | 1 | KILLORGLIN |

Record: 1 of 4 No Filter Search



Notice that no details are displayed for staff numbers 3, 5 and 6.

This is because there is no matching value for StaffNo 3, 5 or 6 in the Properties table.

These staff do not oversee any properties.

To include in the result, the Staff records for which there is **no matching record** found in the Properties table, we must use an **OUTER** Join.

Using an OUTER JOIN will include in the result-set ALL staff records, even if no matching StaffNo value is found in the properties table.



Suppose we wish to list all staff and the properties which they oversee.

```
SELECT S.StaffNo,Surname, Forename, PropNo,Town
FROM Staff S LEFT JOIN Properties P ON S.StaffNo = P.StaffNo;
```



The resulting records are as follows:

Query1

| StaffNo | Surname | Forename | PropNo | Town |
|---|---|---|---|---|
| 1 | SMITH | JOHN | 4 | KILLARNEY |
| 2 | JONES | MARY | 2 | TRALEE |
| 2 | JONES | MARY | 3 | TRALEE |
| 3 | KELLY | MICHAEL | | |
| 4 | KENNY | JOAN | 1 | KILLORGLIN |
| 5 | WOODS | JAMES | | |
| 6 | DUGGAN | ANNE | | |

Record: 1 of 7 No Filter Search

Notice the NULL values in the PropNo field. Any attribute from the Properties table will be NULL.



If we wish to list only those staff who oversee no properties:

```
SELECT S.StaffNo,Surname,Forename,PropNo,Town
FROM Staff S LEFT JOIN Properties P ON S.StaffNo =
                                              P.StaffNo
WHERE Town IS NULL;
```



The resulting records are as follows:

Query1

| StaffNo | Surname | Forename | PropNo | Town |
|---|---|---|---|---|
| 3 | KELLY | MICHAEL | | |
| 5 | WOODS | JAMES | | |
| 6 | DUGGAN | ANNE | | |

Record: 1 of 3 No Filter Search



OR

```
SELECT S.StaffNo,Surname, Forename
FROM Staff S LEFT JOIN Properties P ON S.StaffNo = P.StaffNo
WHERE Town IS NULL;
```

Query1

| StaffNo | Surname | Forename |
|---|---|---|
| 3 | KELLY | MICHAEL |
| 5 | WOODS | JAMES |
| 6 | DUGGAN | ANNE |

Record: 1 of 3 No Filter Search

Oracle have included a special operator in it's command set to facilitate the OUTER JOIN. This operator is **(+)**.

When using this operator in a join, place it with the table which is deficient (lacking) in rows.

```
SELECT *
FROM Staff S, Properties P
WHERE S.StaffNo = P.StaffNo(+);
```

The natural join (inner join) is now an outer join.

The result includes all Staff even if they oversee no properties.