Access/SQL Server - Eliminate Duplicates with SELECT DISTINCT

When you are working with the **SQL SELECT** statement, you will more than likely come across duplicate rows when viewing your query results. This should cause no real problems, and by using the **SQL DISTINCT** keyword in your **SELECT** statement you will be able to remove all duplicate rows from your query results.

The syntax for the **SELECT** Statement including the **DISTINCT** keyword would be:

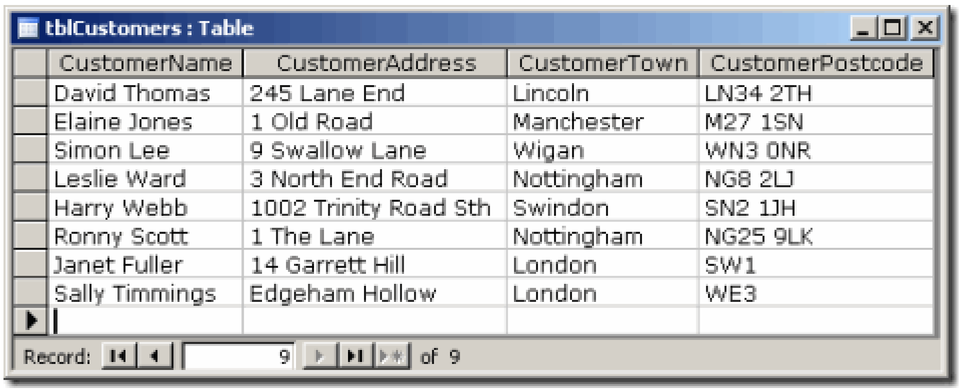
```
SELECT DISTINCT "column_name"

FROM "table_name"
```

DISTINCT is an optional keyword (*see the full list of reserved keywords in **Microsoft Access***) that needs to precede the columns that are specified in the **SELECT** clause. Using **DISTINCT**, the system will evaluate that data contained in all of the columns as a single unit, on a row per row basis, and will eliminate any duplicates that it finds. It will then return the results of the unique rows that remain.

If we take a look at the following example, we will see the difference between running a query for a standard **SELECT** statement, and then running the same statement including the **DISTINCT** keyword.

Our initial database table contains the following *Customer* data:



tblCustomers : Table

CustomerName	CustomerAddress	CustomerTown	CustomerPostcode
David Thomas	245 Lane End	Lincoln	LN34 2TH
Elaine Jones	1 Old Road	Manchester	M27 1SN
Simon Lee	9 Swallow Lane	Wigan	WN3 0NR
Leslie Ward	3 North End Road	Nottingham	NG8 2LJ
Harry Webb	1002 Trinity Road Sth	Swindon	SN2 1JH
Ronny Scott	1 The Lane	Nottingham	NG25 9LK
Janet Fuller	14 Garrett Hill	London	SW1
Sally Timmings	Edgeham Hollow	London	WE3

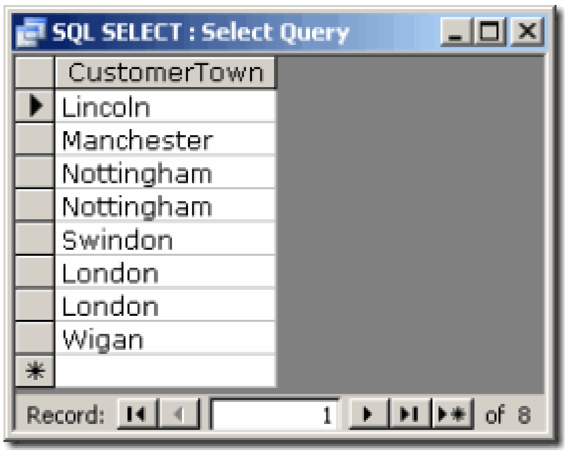
Record: 9 of 9

Our database table, containing **Customer** data.

If we decide to query our data to find out the **Towns** that our **Customers** represent we can run the following **SQL SELECT** statement:

```
SELECT CustomerTown

FROM tblCustomers;
```



CustomerTown
Lincoln
Manchester
Nottingham
Nottingham
Swindon
London
London
Wigan

The results of running the SQL SELECT statement

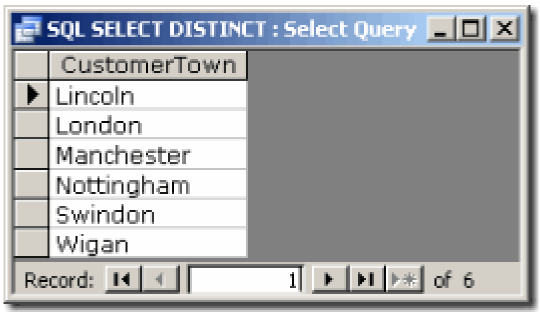
As you will see from the resulting data above, the **SELECT** statement returns every occurrence of a **Town** name included in the **Customers** table. We don't really need to see every occurrence of the Town, so this information is duplicated and unnecessary. If we now run the **SQL SELECT** statement again, however this time we include the optional **DISTINCT** clause, we can eliminate the duplicate information from the resultant data.

The **SQL** for this looks like the following:

```
SELECT DISTINCT CustomerTown

FROM tblCustomers;
```

The resultant data will now display only a single occurrence of each distinct **Town** found in the **Customers** table:



CustomerTown
Lincoln
London
Manchester
Nottingham
Swindon
Wigan

The results of running the **SQL SELECT DISTINCT** statement

Here you see only unique (*distinct*) values contained in the **Town** column of the **Customer** table.