

Boolean Logic

Aims of session:

To enable you to:

- Use Comparison Operators
- Use Logical Operators
- Use on Screen Filters in Access
- Sort data in Access
- Design a compound query in Access
- Describe and use a basic SQL statement

Comparison Operators

Comparison operators determine the relationship between different parts of a search condition, e.g. to search for all students from SHEFFIELD, the condition used may be

HTOWN="SHEFFIELD"

The common comparison operators are shown below.

- = equal to
- <> not equal to
- < less than
- > greater than
- <= less than or equal to
- >= greater than or equal to

Table 4: Comparison Operators

Logical /Conjunctive Operators

Truth Tables

It is often the case that we need to extract sub sets of data from the database, e.g. "give me a list of all students who live in Sheffield". Before we can do this, we need to be aware of the logic that is involved. The following truth tables represent definitions of the logical operators AND, OR and NOT (AND & OR are conjunctive, i.e. they join two tests together).

Columns A and B represent inputs (testable conditions such as "does the student live in Sheffield (Yes or No)). OUT represents



the result, where 1=TRUE and 0=FALSE. An example of a testable condition in the table STUDREC is:

"Does HTOWN=SHEFFIELD"

Another example is:

"Does KIDS=1"

If we wanted to ask for all students who live in Sheffield and who have one child, we need to combine these two testable conditions. How is this done?

Truth Tables provide the theoretical underpinning to this question.

AND

A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2.1: AND truth table

AND is a binary operator which means that it works on two inputs (A & B). It will give a true output if and only if both inputs are true. Combining "Does HTOWN=Sheffield" AND "Does KIDS=1" gives a true result only if the students lives in Sheffield and has one child.

Rule of thumb: AND will narrow down a search.

OR

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

Figure 2.2: OR truth table

OR is a binary operator and gives a true output if either one or both of the inputs is true. Combining "Does HTOWN=Sheffield" OR "Does KIDS=1" gives a true result for all those who live in



Sheffield and all those who have one child regardless of whether they live in Sheffield.

NOT

A	OUT
0	1
1	0

Figure 2.4: NOT truth table

NOT is a unary operator, which means it works on only one input. NOT will give a result of true if the input is false and a result of false if the input is true. Applying NOT to the testable condition "Does HTOWN=Sheffield" will give a list of all students who do not live in Sheffield.

Precedence

These operators (AND, OR & NOT) are evaluated in strict order:

- NOT
- AND
- OR

Venn Diagrams

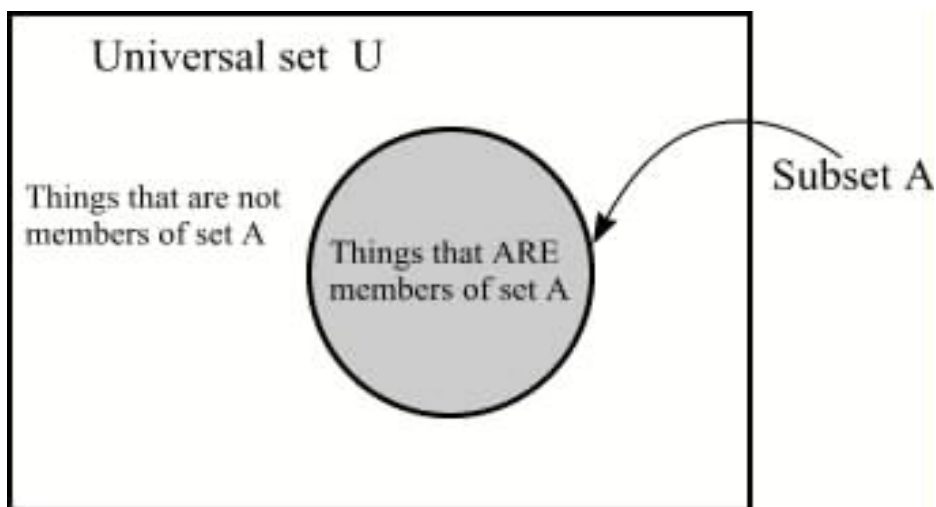


Figure 2.4: The universal Set

Truth Tables and logical operators can also be understood by referring to set theory. Put simply, sets are collections of things and are defined by their membership, for instance the set of all



Northern College Programme students would consist of all of the students who are currently undertaking the Northern College Programme.

A subset is a set that is wholly contained within another set.

The Universal set is the set of all sets. For the purposes of this course the Universal Set is the set of all students (and later all cars, courses etc.).

AND

The logical operator AND can be represented as follows:

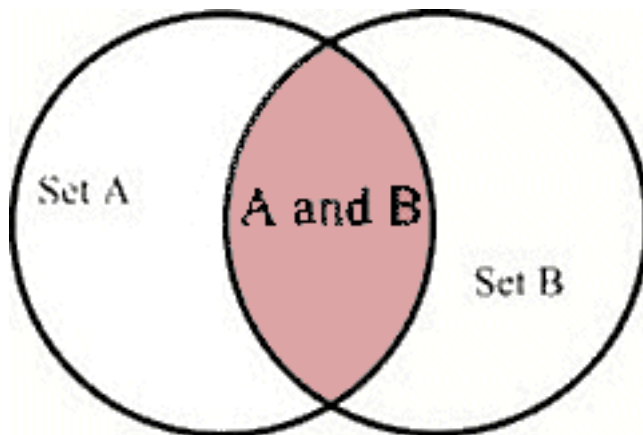


Figure 2.5: A and B

It can be seen that only those things that are members of both set A AND set B are counted when the AND logical operator is used. The AND logical operator narrows down the search.



OR

The OR logical operator can be represented as follows:

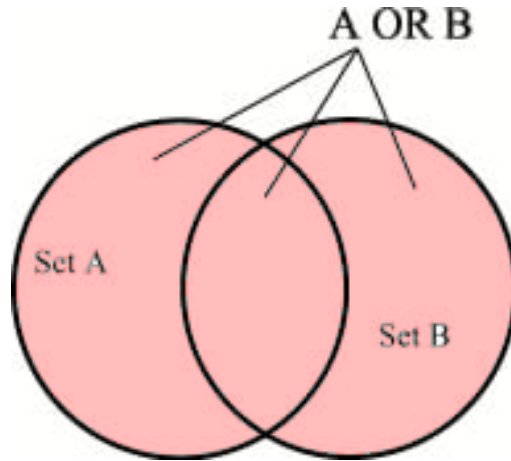


Figure 2.6: A or B

A thing can either be a member of set A, or a member of Set B (or a member of both) to be included by the OR logical operator. Use of the OR logical operator will widen the scope of the search.

NOT

The NOT logical operator includes anything that is not contained by the set in question (that is within the Universal set).

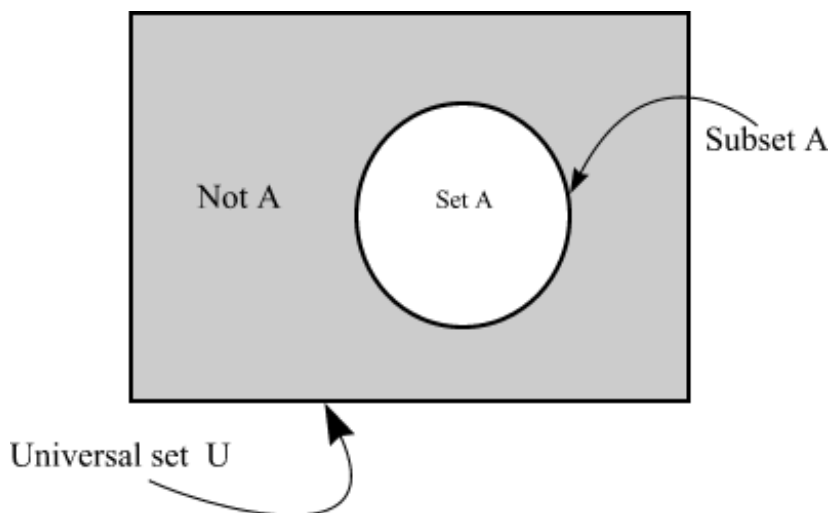


Figure 2.7: NOT A



Queries

Extracting Subsets of Data

Let us suppose that we wish to obtain a list of all of the students in our Studrec table who live in Sheffield. How can we do this?

On Screen Filtering

One easy way of extracting subsets of data from Studrec is to ensure that the table Studrec is on screen in Datasheet mode. Carry out the following steps:

- ☛ Highlight the word "SHEFFIELD" in any field in the HTOWN column.

This is a way of instructing Access to list all records for "HTOWN="SHEFFIELD".

- ☛ Next, select the RECORDS menu and choose the FILTER option.
- ☛ Now choose the FILTER BY SELECTION option (you have already selected HTOWN =SHEFFIELD)

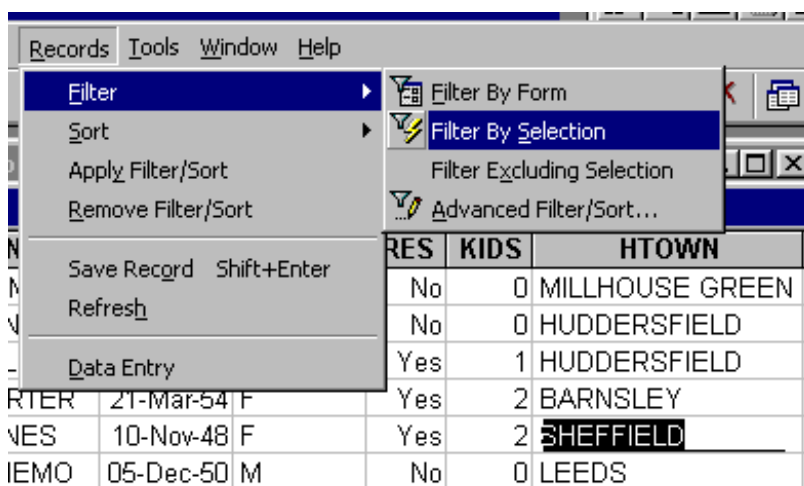


Figure 2.8: Filter by Selection

The Results of this on screen filter should be similar to those shown in Figure 2.9.



Notes

STUDREC : Table									
	SREF	INIT	SNAME	DOB	GENDER	RES	KIDS	HTOWN	Distance
▶	5	A	JONES	10-Nov-48	F	Yes	2	SHEFFIELD	14
	7	K	ARNOTT	01-Aug-60	F	Yes	2	SHEFFIELD	14
	9	N	GREEN	30-Sep-58	M	Yes	1	SHEFFIELD	14
	10	H	JACKSON	21-Apr-41	M	Yes	0	SHEFFIELD	14
	25	P	ARNOTT	30-Jul-65	F	No	1	SHEFFIELD	14
	26	W	PURDUM	01-Sep-69	M	Yes	0	SHEFFIELD	14
	27	B	NADIR	24-Sep-58	M	No	0	SHEFFIELD	14
	28	A	OLIVER	24-Feb-58	F	Yes	1	SHEFFIELD	14
	33	F	NADIR	12-Dec-53	F	Yes	0	SHEFFIELD	14
*						No			

Figure 2.9: Filter Results

This technique is useful only for simple queries, to gain more control we need to examine how we can design our own query files.

Notice that the RECORDS menu also offers you the means of performing on screen sorting of data. This is extremely useful if you wish to list the data in different ways, e.g. to list students by date of birth, and (in another listing) to list students by surname.

Try listing the data by date of birth.

Query Files

An Access database can contain more than tables. Query files are used to define various filter conditions that we may wish to apply to the table, e.g. list all MALE students who live in SHEFFIELD or LEEDS. By defining our query and saving it in a file, we can use it whenever we like without having to redefine the filters.

In this section we are going to work through the example of creating a query that lists only men who live in either Sheffield or Leeds. We are going to list the data fields SREF, INIT, SNAME, GENDER and HTOWN. Finally, we will save the query as "Men from Leeds or Sheffield".



Close the table STUDREC and select the QUERIES tag.



Click on the NEW button, (you will see the following dialogue box - Figure 2.10).



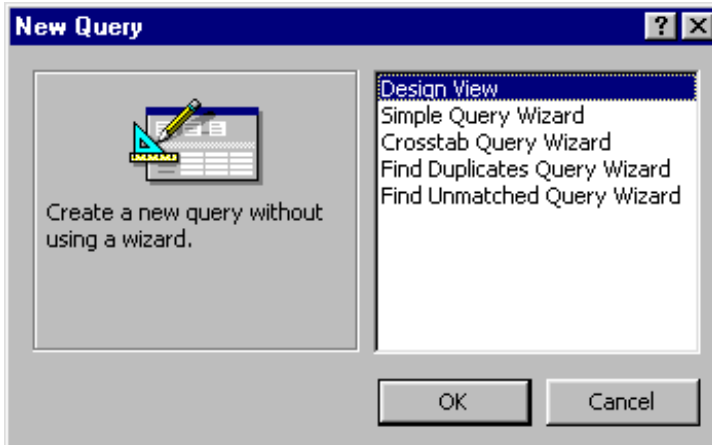


Figure 2.10: New Query Dialogue Box

- ☛ Select the DESIGN VIEW option, click on the OK button. You should see a SHOW TABLE window (Figure 2.11).

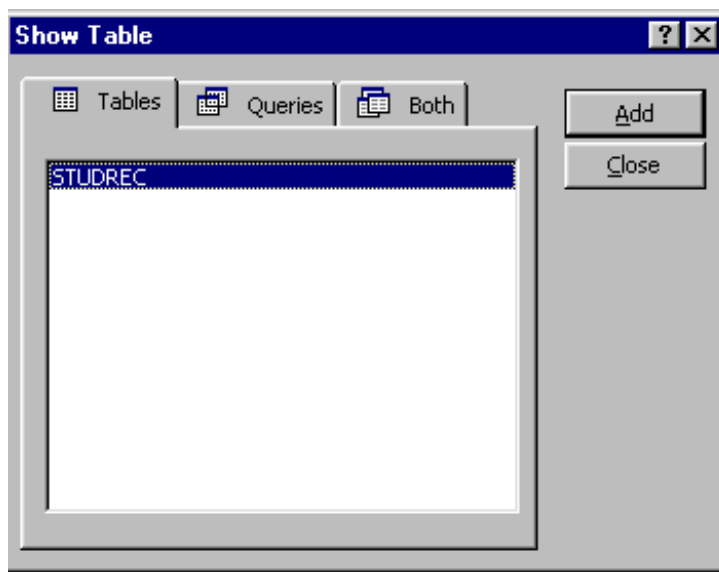


Figure 2.11: Adding a table to a query

- ☛ Select the STUDREC table and click on the ADD button.
- ☛ Click on the CLOSE button, your screen should be similar to Figure 2.12.



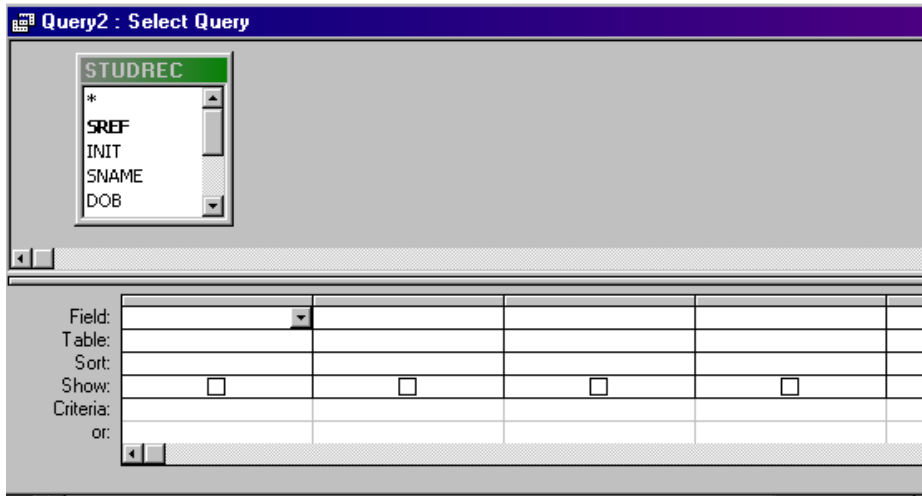



Figure 2.12: The Query Design window

The query design window allows us to define which Columns (fields) and Rows (records) will be filtered out.



First, select which fields are to be listed in which column.

This is done by clicking on the "pull down arrow"  in the field row of each column and selecting the field to be displayed in that column of the query.



Place the following fields in your query

SREF
INIT
SNAME
GENDER
HTOWN

By now, your screen should look like Figure 2.13.



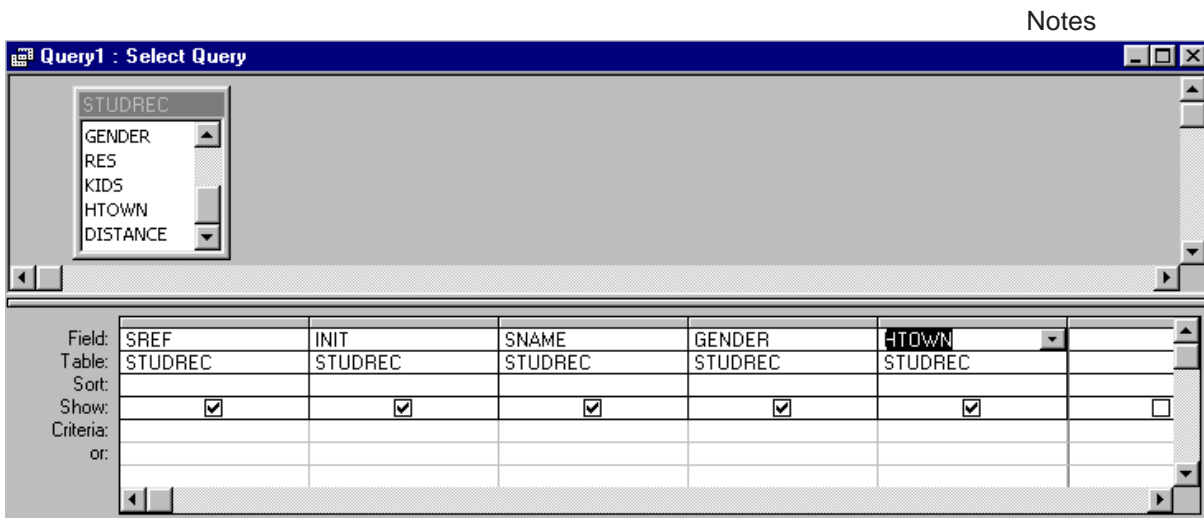


Figure 2.13: Initial Query Design

We now need to tell the query that we wish to list only Men from Sheffield or Leeds. This translates into:

LIST FOR (GENDER="M" AND HTOWN="SHEFFIELD") OR
(GENDER="M" AND HTOWN="LEEDS")

We need to input the selection criteria into the query design form. This is done by putting the first bit (GENDER="M" AND HTOWN="SHEFFIELD") along the first Criteria row of the query design box. The fact that the two criteria are listed along the row means that they are AND'ed.

Next put that second criteria along the second criteria row (the two criteria rows will be OR'ed). The end result will be as shown in Figure 2.14.



Referring to Figure 25, enter the selection criteria as shown



					Notes
Field:	SREF	INIT	SNAME	GENDER	HTOWN
Table:	STUDREC	STUDREC	STUDREC	STUDREC	STUDREC
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				"M"	"LEEDS"
or:				"M"	"SHEFFIELD"

Figure 2.14: Completed selection Criteria for men who live in Leeds or Sheffield



Switch to the Datasheet view to check your work.

You should see only men who live in Sheffield or Leeds as shown in Figure 2.15.

Query2 : Select Query						
	SREF	INIT	SNAME	DOB	GENDER	HTOWN
▶	5	S	ISHAMO	05-Dec-50	M	LEEDS
	9	N	GREEN	30-Sep-58	M	SHEFFIELD
	10	H	JACKSON	21-Apr-41	M	SHEFFIELD
	26	W	PURDUM	01-Sep-69	M	SHEFFIELD
	27	B	NADIR	24-Sep-58	M	SHEFFIELD
	32	E	DAVIS	02-Nov-53	M	LEEDS
*						

Figure 2.15 : Query results



Save your query and call it "Men from Leeds or Sheffield" (FILE/SAVE)

This query will now be ready for you whenever you choose to use it at any time in the future.



Close your query

Using an existing Query.

To use an existing query, simply open Access, load your database, select the query tab and double click on the query file in question.



SQL

SQL stands for Structured Query Language, this is a database management language and is not confined to Access but is used widely throughout the IT industry. It is possible to see what SQL statement has been generated by our work so far. To do this, ensure that the "List all men from Leeds or Sheffield" query is active and

select the **VIEW** menu and choose the **SQL** option.

You should see the following SQL statement on your screen:

```
SELECT STUDREC.INIT, STUDREC.SNAME, STUDREC.DOB,  
STUDREC.GENDER, STUDREC.RES, STUDREC.KIDS,  
STUDREC.HTOWN  
FROM STUDREC  
WHERE (((STUDREC.GENDER)="M") AND  
((STUDREC.HTOWN)="SHEFFIELD")) OR  
(((STUDREC.GENDER)="M") AND  
((STUDREC.HTOWN)="LEEDS"));
```

The SELECT statement defines which fields are to be listed.

The FROM STUDREC bit states which data table the fields are to be found in.

The WHERE bit indicates that there is a filter in action.

The (((STUDREC.GENDER)="M") AND ((STUDREC.HTOWN)="SHEFFIELD")) OR (((STUDREC.GENDER)="M") AND ((STUDREC.HTOWN)="LEEDS")) bit is the filter or search condition. The logic ought to be familiar to you at this stage of the course.

; Note that the entire SQL statement is terminated with a semi-colon.



Using SQL to Generate a Parameter Query

A Parameter Query provides a way of enabling the user to interact with the query. When run, the parameter query will prompt the user to supply information. A PQ could be constructed to list all students from any home town; the user would specify which town that was to be. This technique enables us to devise general queries as opposed to a specific query to list those who live in Sheffield and another for those who live in Silkstone, and yet another for those who live in Barnsley and so on.

More on SQL

SQL comprises of two main sections:

- Data Manipulation Language (DML)
- Data Definition Language (DDL)

Data Manipulation is concerned with Adding, Editing, Deleting and Selecting subsets of data, this corresponds to working with tables in 'datasheet view' in Access. The commands are:

INSERT (add data to a table/s)
UPDATE (edit data)
DELETE (delete data)
SELECT (extract subsets of data)

Data Definition is concerned with creating tables and indexes and corresponds to working in the 'design view' in Access.

The SELECT Statement

The syntax of the SELECT statement can (in its simplest form) be represented as:

SELECT *|<field list> **FROM** <table> [**WHERE** <condition list>];

The terms in **bold type** must be entered exactly as they are shown.

The | symbol indicates a choice (either type a * or a list of field names).

Anything between angled brackets < > must be substituted with specific term/s.

Anything in square brackets [] is optional.



Consider the following SQL statements:

SELECT * FROM studrec;

will list all of the fields (that's what the * wildcard does) and all students from the table studrec.

SELECT init,sname FROM studrec;

will list all students from studrec but only list the fields init and sname.

SELECT * FROM studrec WHERE htown='Huddersfield';

will list all students who live in Huddersfield (but will list all of the fields for each student).

SELECT init, sname,htown FROM studrec WHERE town='Leeds';

will list the fields init, sname and htown and only those students who live in Leeds (from the table studrec).

Note: the * (asterisk) is a wildcard and can be used to stand for any other character string.

LIKE

The LIKE statement in SQL enables us to compare two character strings. This is especially useful when we use the * wildcard. Consider the following example:

SELECT * FROM studrec WHERE htown LIKE 'S*';

This would (at least in Access; there are slight differences in SQL implementations across different platforms) list all students who live in a town that began with S (Silkstone and Sheffield given our test data).

Creating a Parameter Query

We are going to create a parameter query called **any town (P)**. When run, this query will prompt us to type in the first letter (or first few letters if you want to make your selection more specific) of the town that you wish to list students from.



Ensure that you have Access loaded, that you have the Student Records file open and that you have selected the 'Queries' button in the Objects menu (see Figure 2.16).



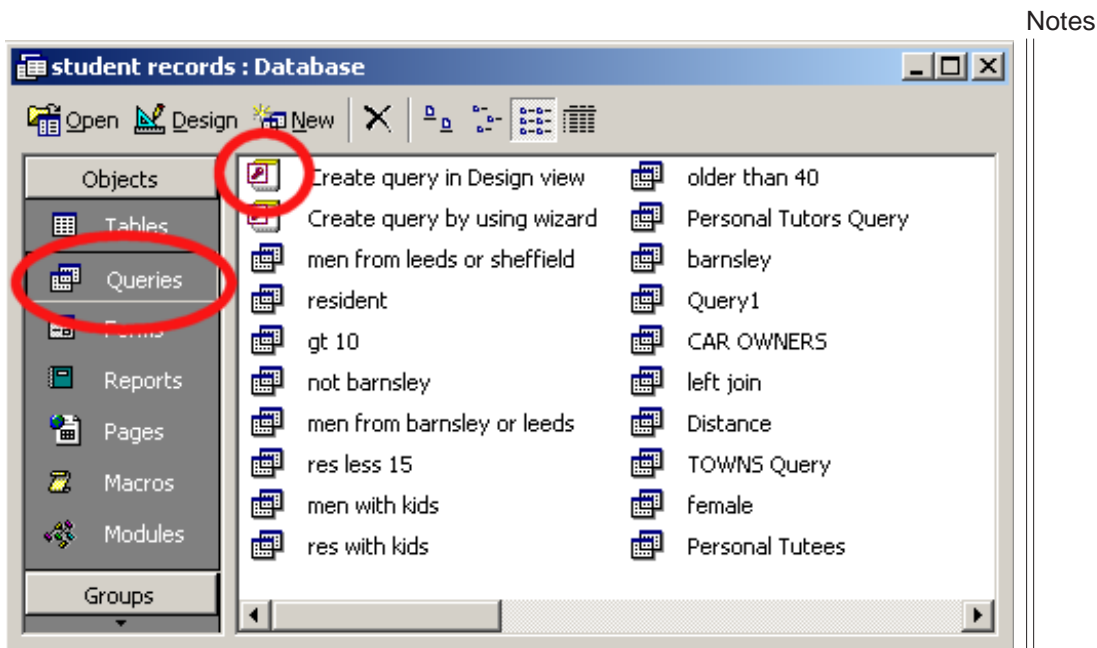


Figure 2.16: Creating a Parameter Query 1



Create a new Query in design view as shown in figure 2.17.

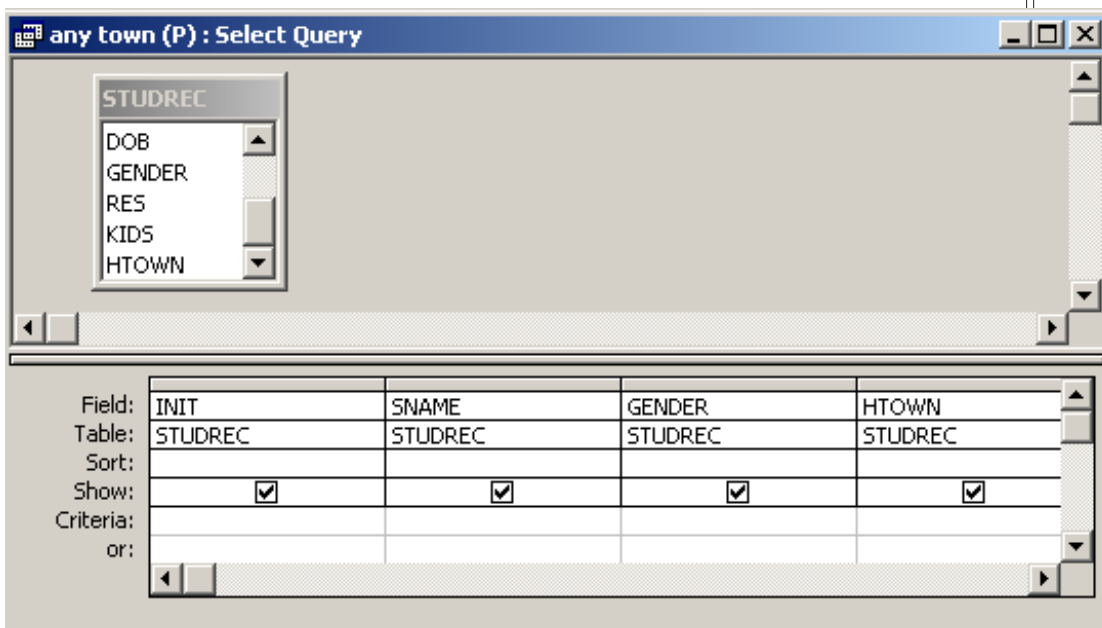


Figure 2.17: Creating a Parameter Query 2

We must now place a condition in the **criteria** section of the **htown** column which will prompt the user to enter the first letter of the town.

The SQL based criteria is:

Like [Which town? (enter first letter):] & ""



Everything that appears between square brackets [] will be displayed by Access as a prompt to the user when the query is run.

Access will replace the prompt with the value of the string that the user typed in response to that prompt, e.g. suppose I typed the letter **S**.

As far as Access is concerned the SQL based criterion looks like this:

Like "S" & "*"

The ampersand symbol (&) is used to concatenate (join together) two separate character strings into one character string. This results in:

Like 'S*'

This will list all those students where the home town begins with the letter S.

Now complete the following steps.



Enter the htown criteria as shown in Figure 2.18.

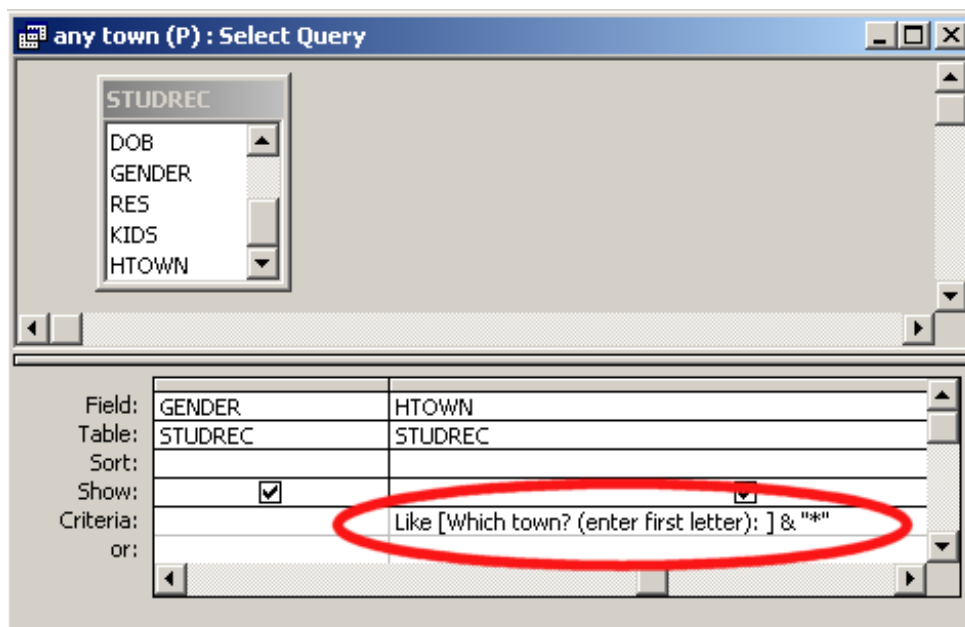


Figure 2.18: Parameter Query 3



Switch to datasheet view and, when prompted, save your query as Any Town (P)



You should now see a prompt as shown in Figure 2.19.

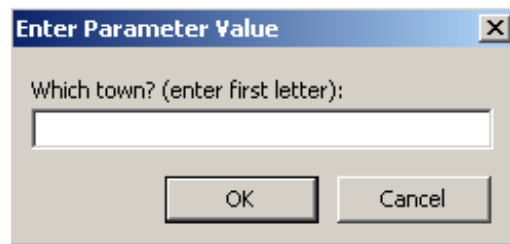


Figure 2.19: Parameter Prompt



Enter the letter S and click on the OK button

You should see a listing similar to Figure 2.20.

	INIT	SNAME	GENDER	HTOWN
▶	A	JONES	F	SHEFFIELD
	K	ARNOTT	F	SHEFFIELD
	N	GREEN	M	SHEFFIELD
	H	JACKSON	M	SHEFFIELD
	N	HEY	F	SILKSTONE
	T	MOSLEY	M	SILKSTONE
	P	ARNOTT	F	SHEFFIELD
	W	PURDUM	M	SHEFFIELD
	B	NADIR	M	SHEFFIELD
	A	OLIVER	F	SHEFFIELD
	T	JAMISON	F	SILKSTONE
	F	NADIR	F	SHEFFIELD
	A	WHITE	M	SILKSTONE
*				

Figure 2.20: Parameter Query 4

Try running this query a few times. What happens when you enter different responses to the prompt?



Exercise 2:

Design query files to answer the following questions. You **MUST** save your queries as detailed in each question.

Unless specifically stated you must list all the data fields in the STUDREC table.

Submit a screencopy of the design view of each query. (this can be done by having WORD open and using the "Print Screen" key to copy the screen to the clipboard. Simply paste to Word, add your name & print. **Please save each answer using the file names given below (this is important as we'll refer to these files later in this course).**

Each question must have your name on it and it must be correctly numbered and titled (e.g. ex. 2.1 List all students from Barnsley)

The deadline is two weeks from receipt of the questions.

Question No.	Question	Save file as
1	List all students from Barnsley	Barnsley
2	List all students who are female	Female
3	List all students who are resident	Resident
4	List all students who live more than 10 miles away	GT 10
5	List all students who are not from Barnsley	Not Barnsley
6	List all male students from Barnsley or Leeds	Men from Barnsley or Leeds
7	List resident students with children (list only INIT, SNAME, RES, KIDS)	Res with Kids
8	List all students who live less than 15 miles away and who are not resident.	Res less 15



			Notes
9	List all men who have children	Men with kids	
10	List all students who are over 40 years old as of 1st August 2001. (list only INIT, SNAME, DOB). Order the data by Date of Birth.	Older than 40	
11	Construct a parameter query to list students by a particular family name. You should prompt the user to enter the first few letters of the name. List the fields init, sname, gender and dob.	By Name (P)	

