MICROSOFT ACCESS GUIDE

Chapter 2 – Relationships and Queries

BTEC Level 3 INFORMATION TECHNOLOGY

Table of Content

1 Defining the relationships 3
1.1 Linking the tables 4
1.2 Setting the links 6
2 Select queries 9
2.1 Planning a query 9
2.2 Using the Query Design 9
2.3 Query 1 finding details of lessons booked on a given day 12
2.4 Selecting fields in Query Design View 13
2.4.1 Query 2 Finding the contact details for students who have not passed the theory test 13
2.5 Query 3 producing a list of instructor's names and addresses 14
2.6 Query 4 Finding lessons between dates 15
2.6.1 Screen Shot evidence 15
2.7 Query 5 Finding morning lessons for a specific instructor between dates 16
2.7.1 Why?…. 16
2.7.2 Query 6 Finding lessons for an instructor on given dates 17
2.7.3 Query 7 Using relational operators 18
2.7.4 Query 8 Using a WildCard 18
2.7.5 Query 9 Using wildcards again 18
2.7.6 Query 10 Logical/Boolean Operators 19
2.8 Parameter queries 20
2.8.1 Query 11 Looking up a Student's details 20
2.8.2 Query 12 Looking up a student's lesson 20
2.8.3 Query 13 Searching for an instructor's lessons by date 21
2.9 Setting up multi-table Queries 22
2.9.1 Query 14 To produce a list of lessons together with student names 22
2.9.2 Query 15 Searching for an instructor's lessons 23
2.9.3 Query 16 Viewing all lessons with full details of instructors and student names 23
2.9.4 View full details of lessons on a certain date – Query 17 23
2.10 Queries using the Totals row 24
2.10.1 Query 18 How many lessons for each Instructor 24
2.10.2 Query 19 Cost by student 25
2.11 Adding a calculated field to a query 26
2.11.1 Query 20 Calculating cost of lessons 26
2.11.2 Query 21 calculating a discount cost 27
3 Index 28

1 Defining the relationships

1. From your teacher, get a copy of "**Activity Sheet 06 Defining Relationships**" and **complete page one only** it before you continue with this booklet.

Unit 2 Activity Sheet 06

Defining Relationships

Unit 2 Creating systems to manage information

Name: _____ Group: _____

Here are the tables from your "Driving School Chapter 2 with Data" database:

tbl Student
StudentID
Title
Surname
Firstname
Address1
Address2
Address3
Address4
PhoneNo
MobileNo
DateOfBirth
Gender
TheoryTestDate
PassedTheoryTest
PracticalTestDate
PassedPracticalTest

TblLesson
LessonNo
StudentID
InstructorID
Date
StartTime
LengthofLesson
CollectionPoint
DropOffPoint
LessonType

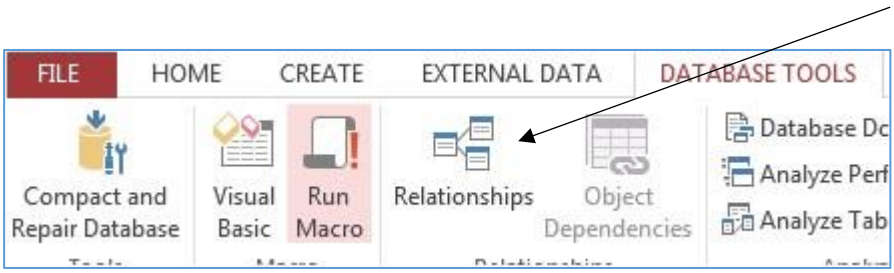
Tbl Instructor
InstructorID
Title
Surname
Firstname
Address1
Address2
Address3
Address4
HomeTelNo
MobileNo
EmailAddress

tblLessonType
LessonType
Cost

1.1 Linking the tables

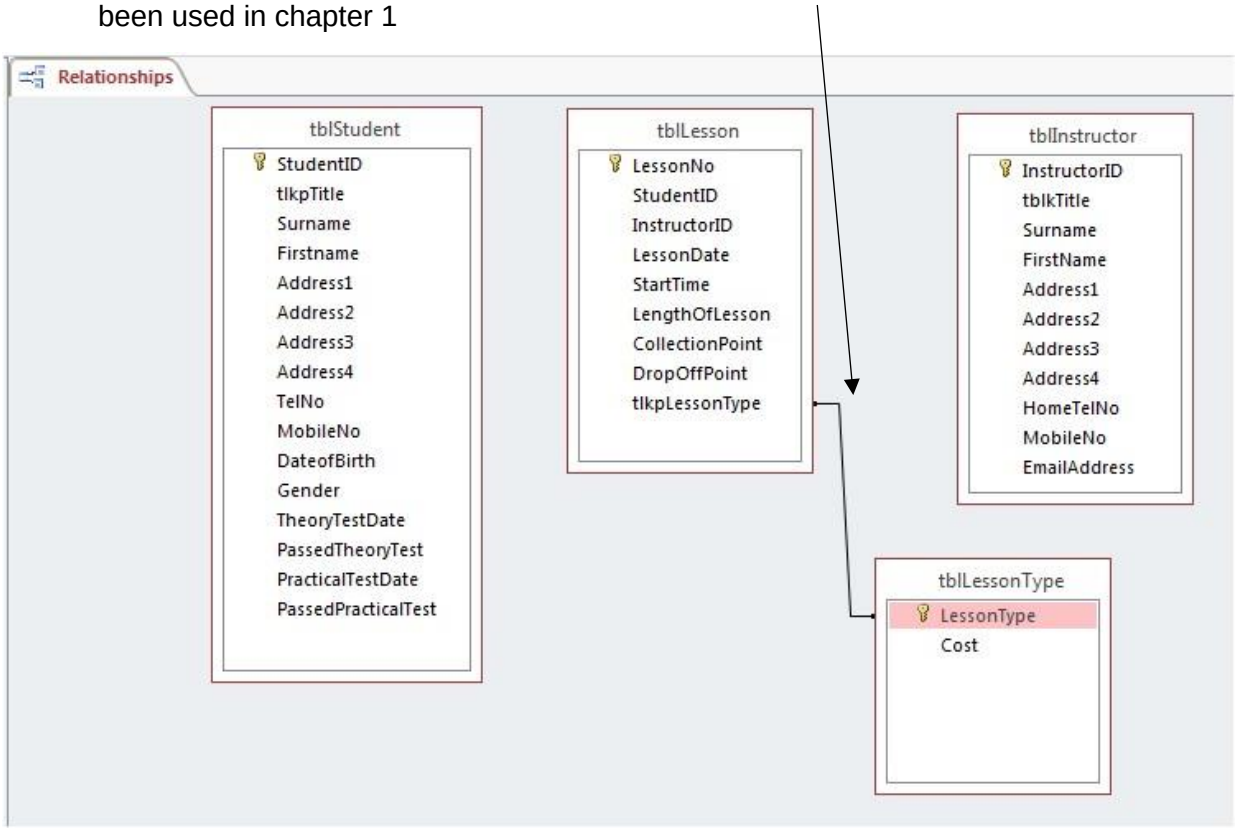
1. Go to the **BTEC IT study directory > Unit 2 > databases and download "Driving school Chapter 2 with data"** and move it into your **BTEC IT folder on your user area**.
2. Open your **"Driving school Chapter 2 with data"**

3. In the Database Window open the **Database Tools** tab and select "**Relationship**" button



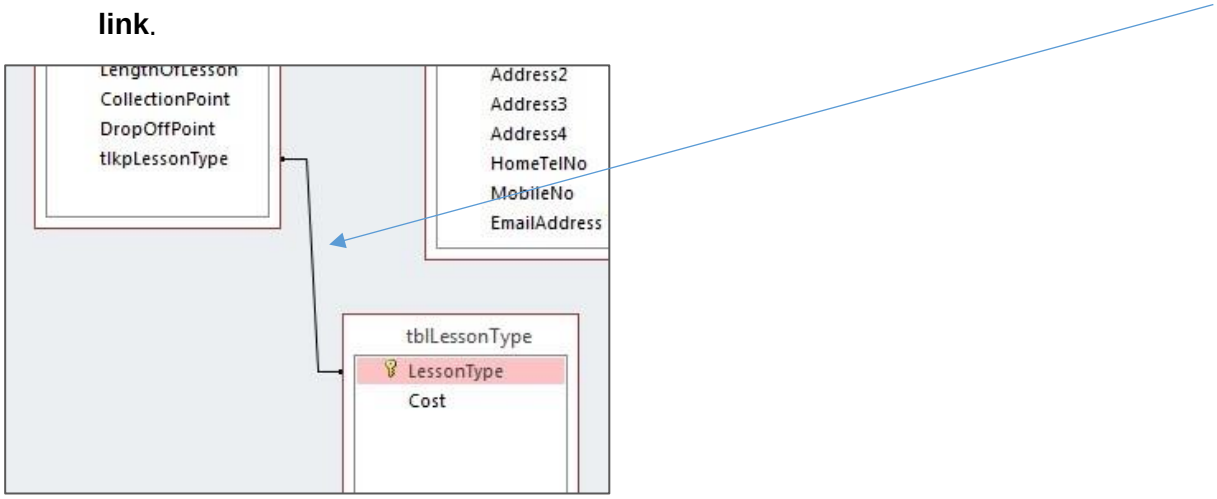
4. Add all four tables to the relationship screen by dragging them in

5. You'll see that one relationship has already been made. This is where a lookup table has been used in chapter 1

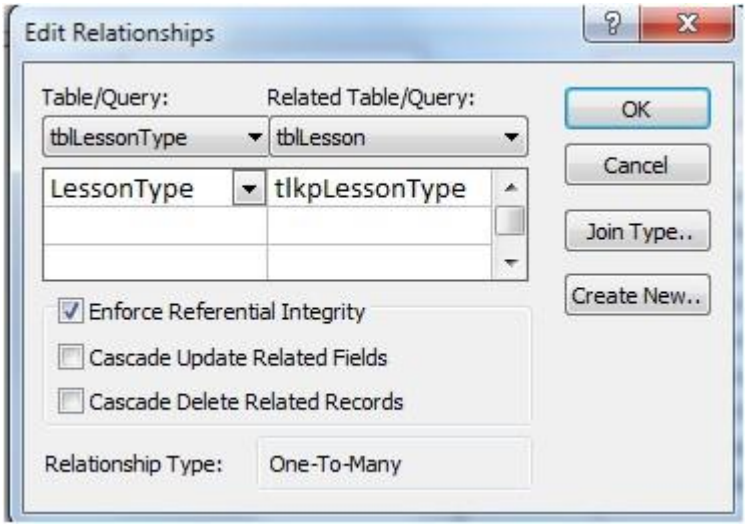


1.2 Setting the links

6. The relationship between the primary key **LessonType** in the **tblLessonType** has already been made with the foreign key **tlkpLessonType** in the **tblLesson**. **Double click on the link**.



7. In the edit Relationship window, **check the Enforce Referential Integrity box**. **DO NOT check** the cascade Delete Related Records box. Click OK



8. Find your "**Activity Sheet 06 Defining Relationships**" page 2, make notes (in your own words) about Referential Integrity:

Referential Integrity

When working with relational databases adding, changing or deleting an item of data in one table may well have an impact on the data in other tables.

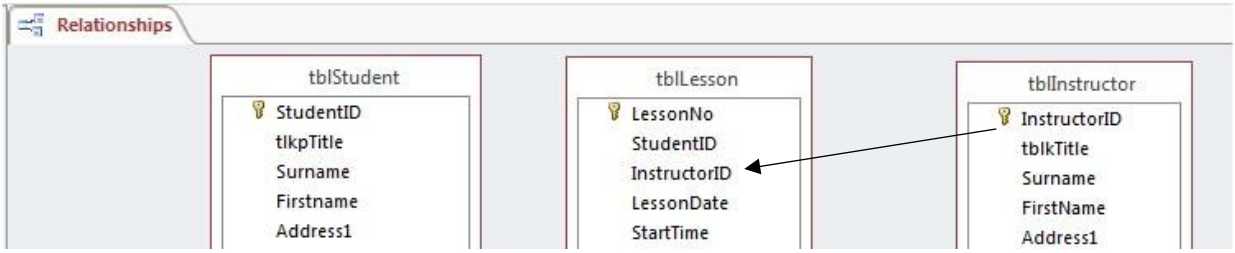
Referential Integrity uses the relationships that have been created through the use of primary and foreign keys to **cascade** any changes.

It ensures that no **orphan** records are created (i.e. records of data that no longer have connections to other data) and **prevents accidental deletion of data** that is related to data in another table.

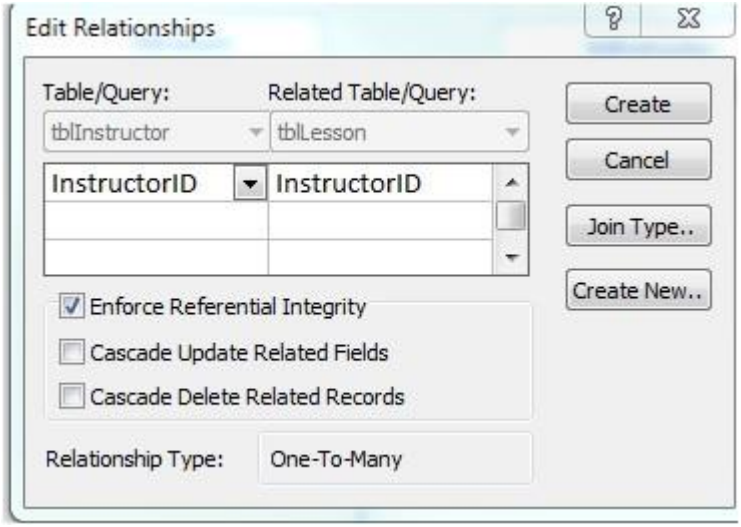
If you want more information, just search the www.

You are now going to set up the rest of the links

9. You are going to make a link between the Instructor table and the lesson table. Click on the primary key **InstructorID** in **tblInstructor** and drag it on top of the foreign key **InstructorID** in **tblLesson**. *The InstructorID can only appear once in tblInstructor but many times in tblLesson.*



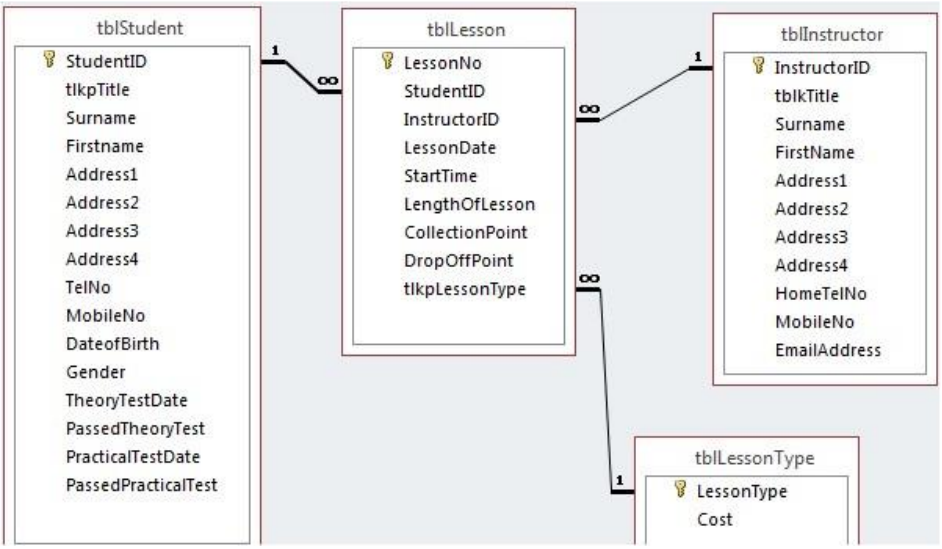
10. When the **Edit Relationship window** appears **check Enforce Referential Integrity** but **DO NO**T check the Cascade Delete Related Records box. **Click on Create.** A link or "relationship line" has been set up between the two tables



Note.

If you *had* checked **Cascade Delete Related Records** and you chose to delete an instructor from tblInstructor then all the lessons for that instructor would be deleted from the Lesson table! Not good!!

11. The **student ID** can appear only one time in the **tblStudent** as it is a **unique ID**. However, a **studentID** can appear many times in **tblLesson** as the **student will need many lessons**. Make a link between the StudentID field in tblStudent and StudentID in tblLesson, making sure the you check "Enforce Referential Integrity".
12. The Relationship window should now look like this:



13. The number **I and the infinity symbol** mean that the relationships are **one-to-many**.
14. Click on the **Save** icon and then close the **Relationships window**.

2 Select queries

In this section you will use queries to search and sort the data in your tables according to certain criteria. Queries provide an easy way of asking questions of your database and producing useful information

2.1 Planning a query

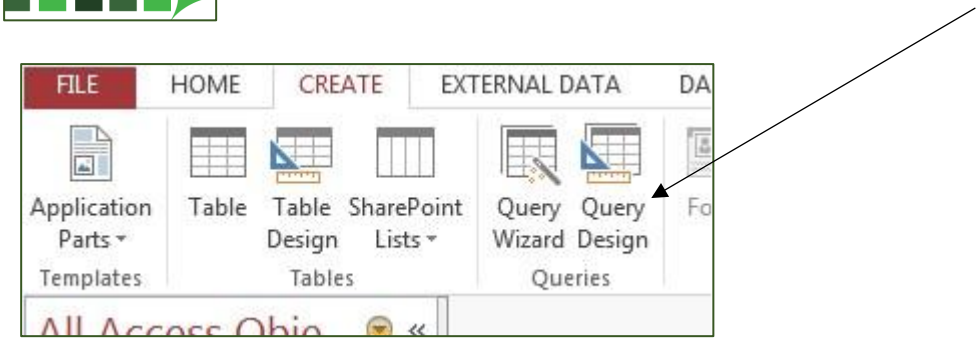
There are usually five steps involved in planning a query

a)Choosing which tables to use
b)Choosing the fields needed in yoru query
c)Setting the criteria to produce the output required
d)Running the query
e)Saving and/or printing the results.

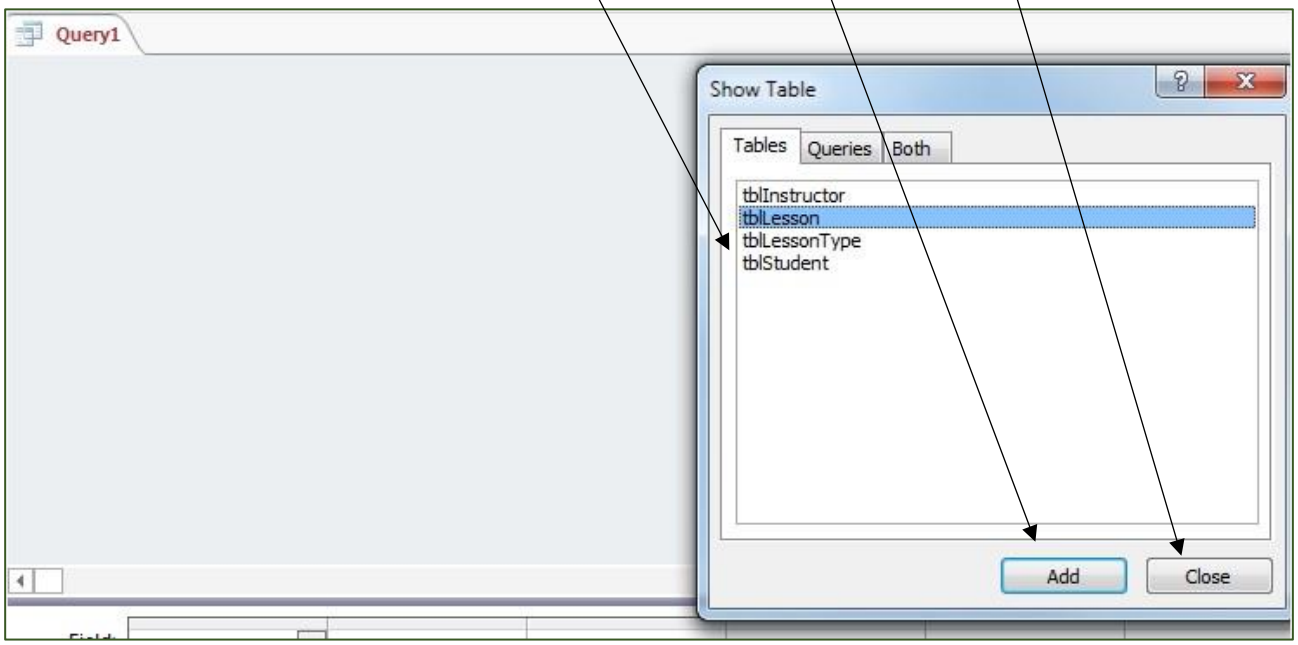
2.2 Using the Query Design

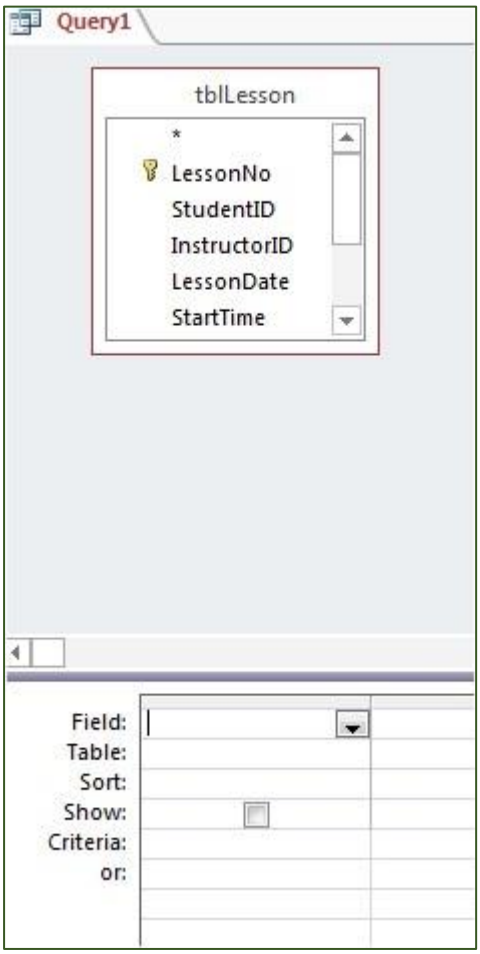
Do This

1. Load the database **"Driving school Chapter 2 with data"**
2. In the Database Window click Create > **Query design**



3. In the Show Table window select **tblLesson**, click on **Add** and then **Close** in the window





4. The Query Design view will now be on your screen. It has **two sections.** The **upper** section contains **the field list** of the table and the **lower** section contains the **QBE (Query by Example) grid** where you design the query. The QBE is made up of 5 rows:

a. **Field** Contains the names of the fields needed for your query

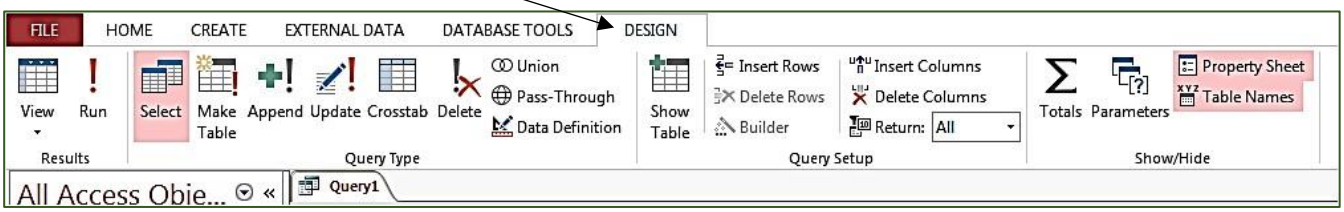
b. **Table** Holds the name of the table containing the selected field

c. **Sort** Offers ascending, descending sort options

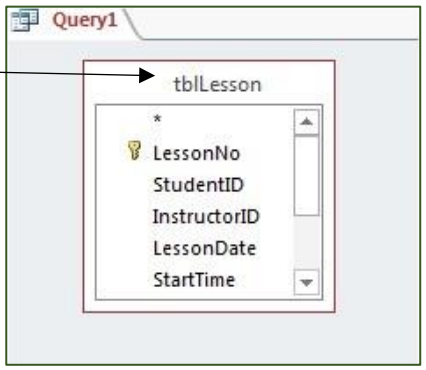
d. **Show** Allows you to hide fields from the output

e. **Criteria** This is where you enter the criteria for yours search

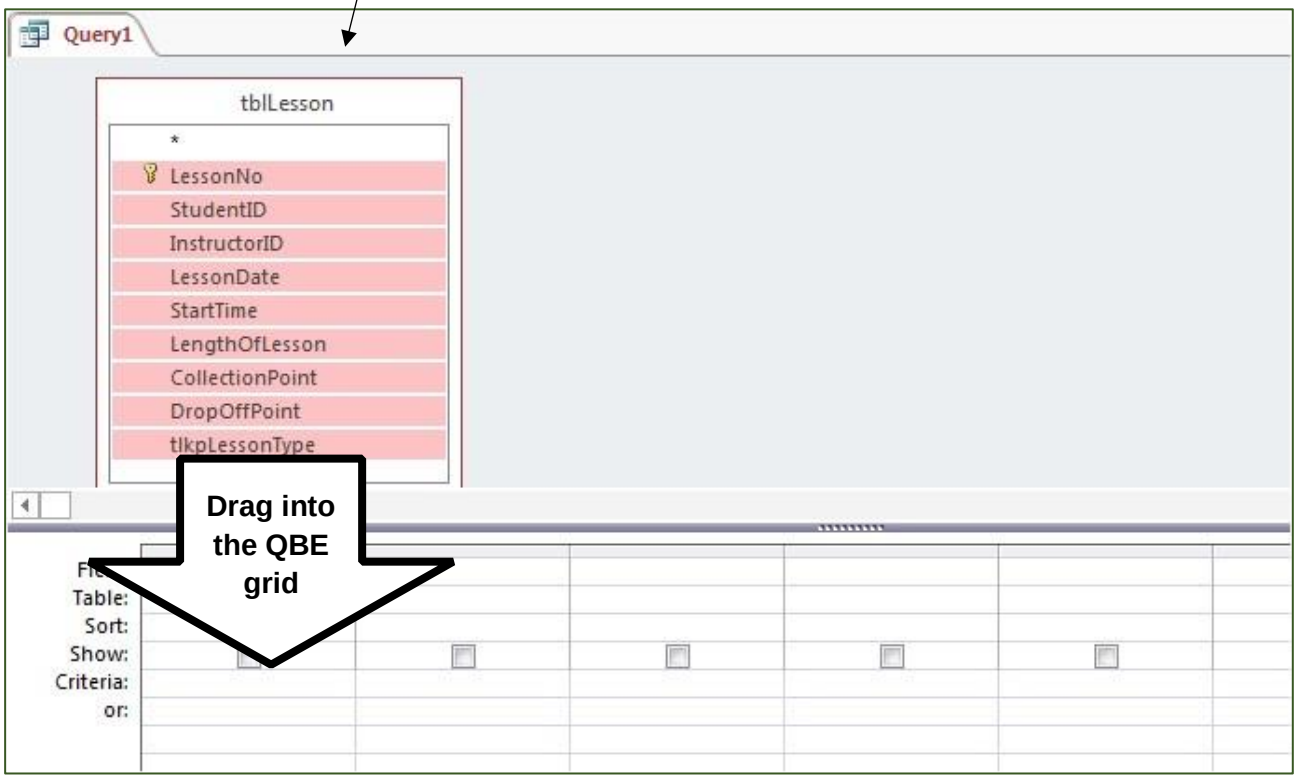
5. The **Query Design toolbar** should be showing (if not, click View > Toolbars > Query Design)



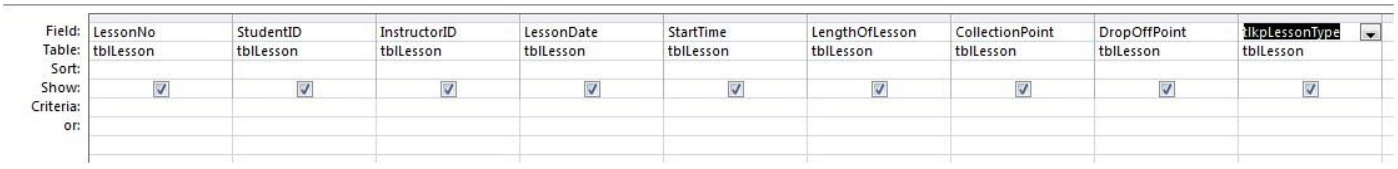
6. For this query you need all the fields to be in the field list. **Double click** on the **tblLesson** title bar of the window



7. When you've done this, all the fields should have a pink highlight. **Drag** the pink highlighted fields down into the QBE grid.



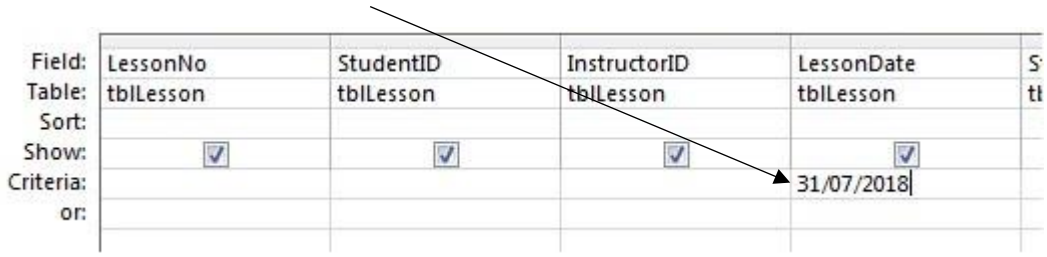
8. The fields should all now appear in the **QBE grid**



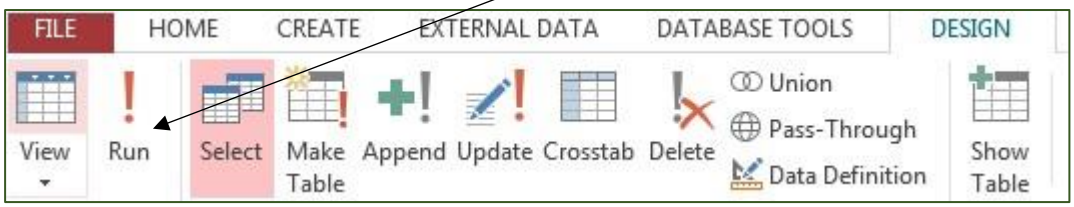
9. You are now ready to start a query.

2.3 Query 1 finding details of lessons booked on a given day

10. You are going to select all the lessons booked on 31/07/17. In the **criteria row** of the **date field** enter **31/07/2018**



11. From the Query Design toolbar, select "**Run**"

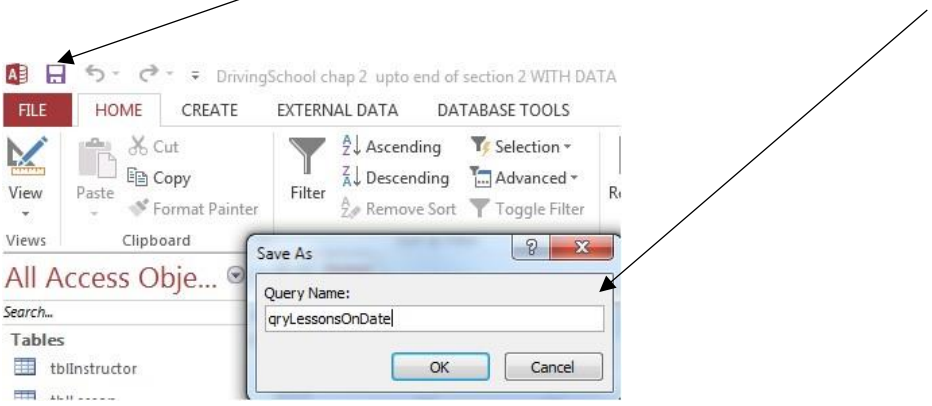


12. There should be ten lessons, like this:

Query1

LessonNo	StudentID	InstructorID	Lesson Date	Start Time	Length of Le	Collection Point	Drop Off Point	Lesson Type
25	10	1	31/07/2018	10:00	1	Barton School	Home Address	Standard
26	12	1	31/07/2018	12:00	1	Home Address	Home Address	Standard
27	2	1	31/07/2018	14:00	1	Home Address	Home Address	Standard
28	13	1	31/07/2018	16:00	1	Home Address	Home Address	Standard
64	25	2	31/07/2018	09:00	1	Home Address	Home Address	Standard
65	20	2	31/07/2018	11:00	1	Home Address	Home Address	Standard
66	26	2	31/07/2018	13:00	2	Home Address	Home Address	Standard
67	15	2	31/07/2018	16:00	1	Home Address	Home Address	Standard
104	31	3	31/07/2018	12:00	1	Home Address	Home Address	Standard
105	34	3	31/07/2018	13:00	1	Home Address	Home Address	Standard
(New)	0	0			1	Home Address	Home Address	

13. Click on the **save** icon and save the query with the name **qryLessonsOnDate**



2.4 Selecting fields in Query Design View

There are a number of ways to select fields from the field list in the query design window:

- In each field cell on the grid is a drop-down list from which fields can be chosen
- Double click on the title bar in the Field List table. This highlights all the field names.
- Click on any one of the of the fields in the Field list and drag them down individually

2.4.1 Query 2 Finding the contact details for students who have not passed the theory test

1. In the Database window, select Create > Query Design
2. In the Show Table window select **tblStudent**, click on Add and then close
3. Select the following fields by dragging them into the QBE grid:
   a. Firstname
   b. Surname
   c. MobileNo
   d. PassedTheoryTest
4. We are trying to find all students who have not yet passed their theory test. The PassedTheoryTest field is a Yes/No data type, so type **No** in the **criteria row** of the **PassedTheoryTest field**.

Field:	Firstname	Surname	MobileNo	PassedTheoryTest
Table:	tblStudent	tblStudent	tblStudent	tblStudent
Sort:				
Show:	☑	☑	☑	☑
Criteria:				No
or:				

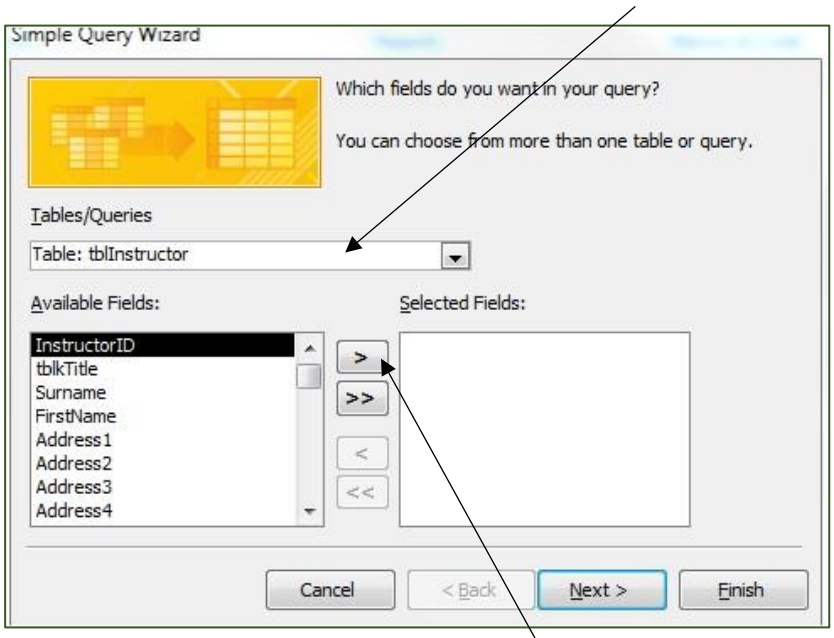
5. Click on the **Run** button and there should be **seven results**.
6. Save the query with the name **qryNotPassTheory**

Note: if you need to get back to the Design view of the query, simply click on the **Design view button** in the tool bar (home tab)

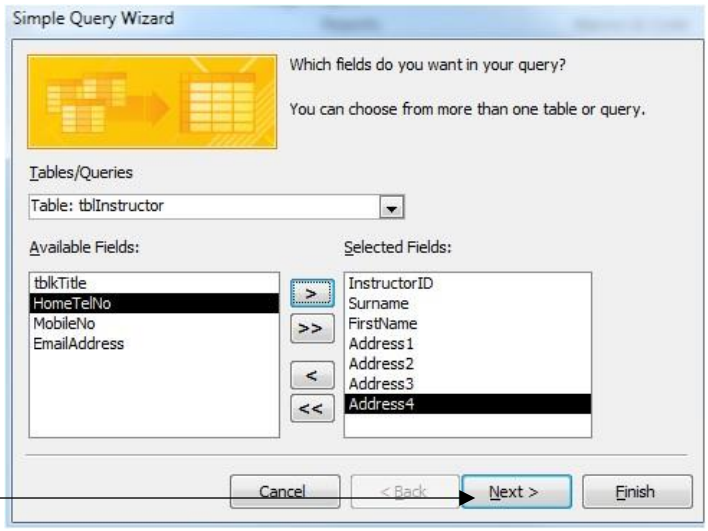
2.5 Query 3 producing a list of instructor's names and addresses

You will be using the Query Design for the majority of your queries but there are other ways of creating simple queries.

1. Click on Create > Query Wizard > simple Query Wizard and click OK 2. Select **tblInstructor** from the **Tables/queries drop down**



3. Select the following fields by using the single



arrow:

a. InstructorID
b. Surname
c. Firstname
d. Address1

e. Address2
f. Address3
g. Address4

4. Click Next
5. Name the query **qryInstructorAddresses** and click **Finish**

2.6 Query 4 Finding lessons between dates

You can select a range of record using relational operators <, <, <=, >=, <> and the Logical/Boolean operators NOT, BETWEEN, OR and AND

Suppose you with to view lessons between certain dates or to print a list of lessons for the coming week

1. Create a query using **Query Design**
2. From the Show table window select **tblLesson** – click on Add then close.
3. Add the fields LessonNo StudentID, InstructorID, LessonDate and CollectionPoint
4. You are going to select all the lessons between 30th July and 2nd Aug
5. In the criteria row of the **LessonDate** column enter **>=30/07/2018 AND <=02/08/2018**

6.

7.

AND 02/08/2018 and click on **Run** again

8. You'll find that you get the same results…there are often more than one way to write queries

Field:	LessonNo	StudentID	InstructorID	LessonDate	CollectionPoint
Table:	tblLesson	tblLesson	tblLesson	tblLesson	tblLesson
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:				>=#30/07/2018# And <=#02/08/2018#	
or:					

Field:	LessonNo	StudentID	InstructorID	LessonDate	CollectionPoint
Table:	tblLesson	tblLesson	tblLesson	tblLesson	tblLesson
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:				Between #30/07/2018# And #02/08/2018#	
or:					

Run the query and you'll see 32 records

Go back to **Design view** and change the criteria to read **BETWEEN 30/07/2018**

9. Save the query with the name **qryBetweenDates**

2.6.1 Screen Shot evidence

For your assessment you will produce screenshot evidence of queries (and other components). Be aware that you need to make sure that all screenshots that you produce **must show the whole criteria** of your queries (see the two screenshots above) and if any of the **criteria is cut off (truncated) you will not** gain marks for that query.

2.7 Query 5 Finding morning lessons for a specific instructor between dates

It is possible to create queries **using criteria in more than one field**. In the next query you are going to select a specific instructor's lessons between certain dates

1. Create a query using **Query Design,** adding the **tblLesson**
2. Put **all** the fields into the QBE grid
3. In the criteria row of the **InstructorID** field type **3** (this will only select Instructor number 3)
4. Click the **Run** button and you'll see all the lessons for Instructor no 3
5. Go back to the **Design view** and **type in BETWEEN 05/08/2018 and 07/08/2018** in the criteria row of the **LessonDate** field and click on the **Run** button again. You should now have 8 records
6. Go back to the **Design View** of the query and type **<=12:00** in the S**tartTime** field (this will find all the morning lessons that start at or before 12:00) and Run the query again.
7. You should have 6 records displayed
8. Save the query as **qryMorningLessonsInstructorThree**

Field:	LessonNo	StudentID	InstructorID	LessonDate	StartTime	LengthOfLesson	Collecti
Table:	tblLesson	tblLesson	tblLesson	tblLesson	tblLesson	tblLesson	tblLess
Sort:							
Show:	☑	☑	☑	☑	☑	☑	
Criteria:			3	Between #05/08/2018# And #07/08/2018#	<=#12:00:00#		
or:							

LessonNo	StudentID	InstructorID	Lesson Date	Start Time	Length of Le	Collection Point	Drop Off Point	Lesson Typ
110	36	3	05/08/2018	09:00	1	Home Address	Home Address	Standard
111	41	3	05/08/2018	11:00	1	Home Address	Home Address	Standard
112	40	3	05/08/2018	12:00	1	Home Address	Home Address	Standard
114	38	3	06/08/2018	10:00	1	Home Address	Home Address	Standard
115	36	3	07/08/2018	10:00	1	Home Address	Home Address	Standard
116	31	3	07/08/2018	12:00	1	Home Address	Home Address	Standard

2.7.1 Why?....

In the last activity you clicked on the Run button each time you added more criteria, why do you think I asked you to do this?

Have a think about it and then ask your teacher to explain

2.7.2 Query 6 Finding lessons for an instructor on given dates

1. Create a new query using Query Design
2. Add the tblLesson table to your query (Add then Close)
3. Add the following fields to your query StudentID, InstructorID, LessonDate, StartTime, CollectionPoint and Lesson Type
4. Add criteria to find the instructor with the ID of 2 – Run the query
5. Go back to **Design View** and enter the criteria to find for that instructor on the 31/07/2018 and **Run** the query.

You now can see that it is easy to specify criteria in more than one field. This is called an AND search.

You may also wish to look for records which meet one criterion OR another. For example you may wish to view lessons on one day OR another. This is sometimes called an OR search.

6. Using the same query as in step 5 above, go back to Design view
7. You now going to add another date, this time using the "or:" row and Run the query

Field:	StudentID	InstructorID	LessonDate
Table:	tblLesson	tblLesson	tblLesson
Sort:			
Show:	☑	☑	☑
Criteria:		2	#31/07/2018#
or:			#08/08/2018#

8. When you run the query you'll notice that you now get lessons with another instructor, go back to Design View an alter the query so that only instructor number 2's lessons are selected. Run the query to check your results.
9. Save the query with the name **qryInstructor2ORdates**

StudentID	InstructorID	Lesson Date	Start Time	Collection Point
25	2	31/07/2018	09:00	Home Address
20	2	31/07/2018	11:00	Home Address
26	2	31/07/2018	13:00	Home Address
15	2	31/07/2018	16:00	Home Address
22	2	08/08/2018	10:00	Home Address
20	2	08/08/2018	13:00	Home Address
29	2	08/08/2018	15:00	Home Address
25	2	08/08/2018	18:00	Home Address
0	0			Home Address

Your teacher will now set you more queries to practice with

2.7.3 Query 7 Using relational operators

Relational operators in Access include:

< **Less than**

<= **Less than and equals to**

> **Greater than**

>= **Greater than and equals to**

1. Create a new query using Query Design and add **tblLesson** then add all fields to the grid
2. You are going to find all lessons that are booked in before 08/08/2018. In the criteria cell of the **LessonDate field type <04/08/2018**
3. Run the query.

4. Now go back to Design view and change the operator so that the query selects all lessons after 04/08/2017. Run the query and see if its worked
5. Go back to Design view and this time alter the criteria so it selects all lessons **on or before 04/08/2018.** Run the query and you should find 68 records.
6. Save the query as **qryOnOrBefore**

2.7.4 Query 8 Using a WildCard

Wildcards are special characters that can stand in for unknown characters in a text value and are handy for locating multiple items with similar, but not identical data.

One of the characters that you can use for wildcards is the asterisk (*)

1. Create a new query using Query Design and add **tblLesson** then add all fields to the grid
2. You are going to select all test driving lessons that are booked for July.
3. In the criteria cell for the LessonDate field type ***/07/2018** (this will find any day in the month of July). **Run** the query and only dates in July should appear.
4. Go back to de**sign view** and type **Test** in the **criteria** cell of the (tlkpLessonType) **LessonType** field. **Run** the query and you should see the two test lessons that are booked for July. Save the query as **qryJulyTestLessons.**

2.7.5 Query 9 Using wildcards again

1. Create a new query using Query Design and add **tblStudent** then add **all** fields to the grid
2. One of the driving instructors is having trouble remembering a student's name. They know it begins with the letter K but can't remember the name.
3. In the criteria cell of the **Firstname** field type **K*** (this will find all the names that start with K but end in anything). **Run** the query
4. You should see three records – Kathryn, Kelly and Kirsty. Save the query **qryKFirstName**

2.7.6 Query 10 Logical/Boolean Operators

Logical operators are sometimes known as Boolean operators; these are the ones you need to know:

AND

OR

NOT

BETWEEN …. AND…..

1. Create a new query using Query Design, add **tblLesson** and put **all** the fields in the grid
2. In this query you are going to select all the records that are for introductory or test lessons. In the criteria cell of the **tlkpLessonType** field type **introductory OR test**

3. Run the query to see the results.
4. Go back to Design View and **delete** the **criteria that you just typed into the Lesson type field.**
5. Now add criteria into the **StartTime** field that will select all lesson that start at either **11:00** or **12:00.** Run the query and you should see 27 records
6. Go back to Design View and in the criteria cell of the **LessonDate** field type in **BETWEEN 04/08/2018 AND 06/08/2018**. You should now see that only lessons between those dates are showing.
7. Finally, go back to Design View and add the criteria in the LessonType field that will remove Introductory lessons from the query (hint: use the NOT operator!). Run the query again and you should see these results:

LessonNo	StudentID	InstructorID	Lesson Date	Start Time	Length of Le	Collection Point	Drop Off Point	Lesson Type
33	6	1	04/08/2018	11:00	1	Home Address	Home Address	Standard
37	11	1	05/08/2018	11:00	1	Home Address	Home Address	Standard
41	12	1	06/08/2018	12:00	1	Home Address	Home Address	Standard
73	15	2	04/08/2018	11:00	1	Home Address	Home Address	Standard
77	28	2	05/08/2018	12:00	1	Home Address	Home Address	Standard
80	20	2	06/08/2018	11:00	1	Home Address	Home Address	Standard
81	25	2	06/08/2018	12:00	1	Home Address	Home Address	Standard
111	41	3	05/08/2018	11:00	1	Home Address	Home Address	Standard
112	40	3	05/08/2018	12:00	1	Home Address	Home Address	Standard

8. Save the query as **qryBetweenOrNot**

2.8 Parameter queries

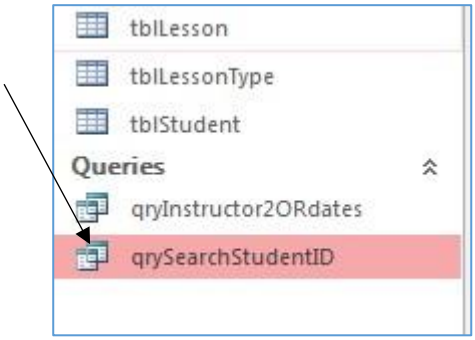
Up until now you've been practicing Select Queries which are very useful if you have to run the query frequently and use different criteria each time. In this section you will learn how to use **Parameter** queries.

Parameter queries overcome the problem by allowing you to enter the criteria each time the query is run. On running the query a dialogue box will appear asking you to enter the details.

2.8.1 Query 11 Looking up a Student's details

1. Create a new query using Query Design and add the tblStudent table.
2. Select the following fields and add them to the QBE grid – StudentID, Surname, Firstname, Address1, Address2, Address3, Address4, MobileNo, DateOfBirth, Gender.
3. In the criteria cell for StudentID type **[Enter the ID number]** (please note that the square brackets are required.
4. Run the query and enter **2** in the dialogue box and student 2's details will appear.
5. Save the query as **qrySearchStudentID** and close it.

6. Open the **qrySearchStudentID** and the parameter window will open. Type in the number of any student to see their details



2.8.2 Query 12 Looking up a student's lesson

7. Create a new query using Query Design and add the **tblLesson** table
8. Select the following fields to appear in the QBE grid**: StudentID, LessonDate, StartTime and LengthOfLesson**
9. Set up a parameter query in the StudentID field that asks the user to enter the StudentID number (remember that square brackets are needed)
10. Run the query and enter **5** when prompted and you'll see all lessons that student 5 has booked.
11. Save the query **qryStudentLesson**

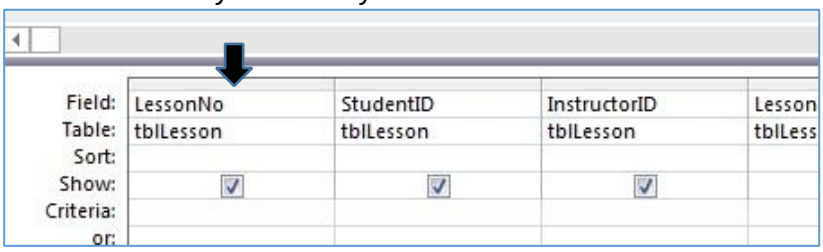
2.8.3 Query 13 Searching for an instructor's lessons by date

In this activity you are going to set up a query with two parameters:

1. Create a new query using Query Design and add **tblLesson**
2. Add **all** the fields to the QBE grid

Deleting fields from the grid.

3. Hover your mouse just over the **LessonNo** field in the grid and you'll see it change into a small black arrow. Click your mouse and the field will be highlighted in black. Press the delete key on the keyboard to delete that field.



4. Delete these fields in the same way: **DropOffPoint** and **LessonType**
5. Now, in the criteria cell for the **InstructorID** field please type in **[Please enter Instructor's ID]**
6. In the criteria cell for the **LessonDate** field please type in **[Enter the date]**
7. Run the query and you'll be asked for an Instructor's ID – Please enter **1** click OK
8. When you are asked for the date, please enter **31/07/2018**, click **OK** and you'll see the lessons that have been booked for instructor 1 on that date
9. Save the query as **qryInstructorLessonDate**

2.9 Setting up multi-table Queries

Earlier in these activities you designed four tables: tblStudent, tblInstructor, tblLesson and tblLessonType and you created the relationships between those tables

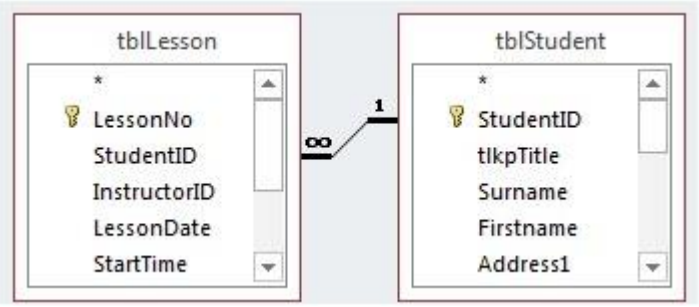
Open the Driving school database and have another look at the relationships that you created

In this section you're going to learn how to base your queries on more than one table and start to use4 the relationship you have set up. In addition you will see how you can use queries to do calculations

2.9.1 Query 14 To produce a list of lessons together with student names

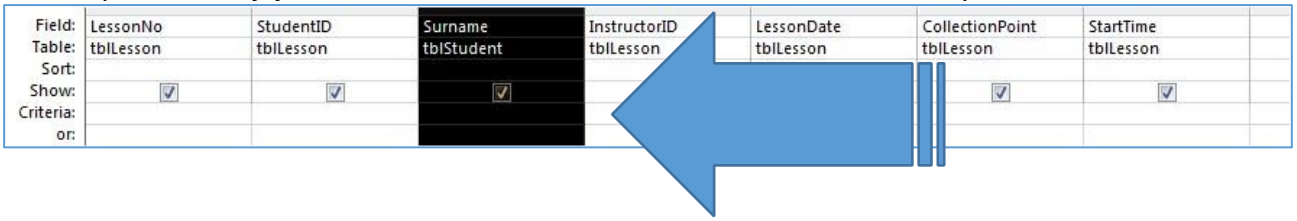
Do This

Design



1. With the **Driving school database open**, create a query using **Query 2.** From the Show Table window add both **tblLesson** AND **tblStudent**

3. From **tblLesson** drag and drop fields **LessonNo**, **StudentID**, **InstructorID**, **LessonDate**, **CollectionPoint** and **StartTime** into the grid
4. From **tblStudent** drag and drop **Surname** field so that it appears to the right of **StudentID.** (Alternatively you could have entered the fields in the order shown!)



5. Run the query and save it as **qryLessonAndNames**

2.9.2 Query 15 Searching for an instructor's lessons

1. Using Query Design create a new query and add both **tblLesson and tblInstructor**
2. From tblLesson drag these fields into the grid: **LessonNo, StudentID, InstructorID, LessonDate, StartTime**
3. From **tblInstructor** drag these fields into the grid: **Surname**…drag this to the right of InstructorID
4. In the **criteria** cell of the **InstructorID** type a parameter query **[Enter Instructor ID]**
5. **Run** the query and type **2** into the dialog box. The query produces a list of lessons for Instructor ID 2!
6. Save the query as **qryInstructorLessons**

2.9.3 Query 16 Viewing all lessons with full details of instructors and student names

1. When you create the new query, this time add **tblInstructor, tblLesson and tblStudent** please.

2. From **tblLesson** add **all the fields to the grid**. From **tblInstructor** add **Surname** and **Firstname**. From tblStudent add **Surname** and **Firstname**
3. In design view drag and drop the Instructor's Surname and firstname to the right of InstructorID and do the same for the student's name
4. Save the query as **qryFullDetails**

The next two queries are going to be used later on in your Access training

2.9.4 View full details of lessons on a certain date – Query 17

1. Open the qryFullDetails, in the criteria row of the LessonDate column type **[Please enter the Lesson Date]**

2. Use **File**, **Save As, Save Object As** to save the query with a **new** name **qryFullDetailsByDate.**
3. **Run** the query and you'll be prompted for a date. Access will display full details of the lessons on that date along with the names or the students and instructors

2.10 Queries using the Totals row

An often-overlooked feature of the ordinary select query is its ability to **calculate totals**. Supposing the data you wish to analyse is contained in just two fields? The answer is to create a select query and make use of the totals option. Totals can do more than just add up too! Here's how it works…

2.10.1 Query 18 How many lessons for each Instructor

1. Create a new query using Query Design and add **tblLesson**, **tblLessonType**, **tblInstuctor**
2. Add these fields to the grid in this order**: InstructorID from tblLesson**, **Surname and Firstname from tbl Instructor** and **LessonNo from tblLesson**

3. Click on the big Sigma "**Totals**" icon in the toolbar and you'll notice a new row appear in the query grid "Total"

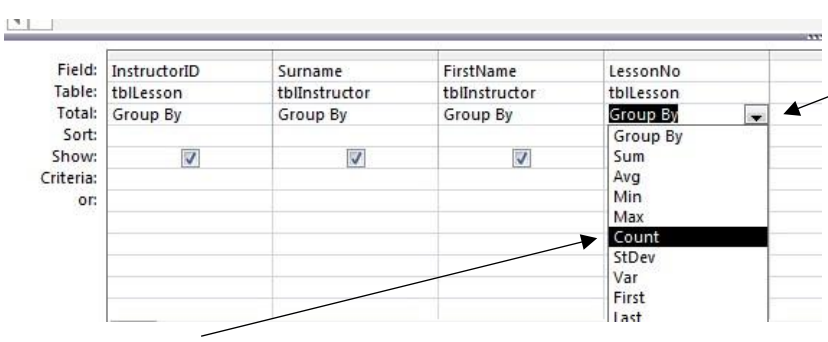
Before you click on the "totals" icon

Field:	InstructorID	Surname	FirstName	LessonNo
Table:	tblLesson	tblInstructor	tblInstructor	tblLesson
Sort:				
Show:	☑	☑	☑	☑
Criteria:				

After you click on the "totals" icon

Field:	InstructorID	Surname	FirstName
Table:	tblLesson	tblInstructor	tblInstructor
Total:	Group By	Group By	Group By
Sort:			
Show:	☑	☑	☑
Criteria:			

4. In the **Total** row of the LessonNo field click on the down-arrow next to **Group by**



5. And select **Count**.
6. Run the query and you'll see a count of how many lessons each instructor has booked. Save is as **qryCountOfLessons**

2.10.2 Query 19 Cost by student

In this query you are going to use the totals row again:

1. Create a query using **Query Design** and add **tblStudent, tblLesson and tblLessonType**
2. From **tblStudent** add **Surname** and **FirstName** and from **tblLessonType** add **Cost** to the grid
3. If the total row isn't showing click on the Big Sigma "totals" icon.
4. In the **total row of the Cost field** use the **drop down** list to select **Max** (the MAX function calculates the highest value). When you Run the query you'll see the maximum amount each student has paid for a lesson.
5. Go back to Design View and change the totals row of the Cost column so a students average cost will be displayed. Run the query.
6. Name the query, **qryAverageCostPerStudent**

2.11 Adding a calculated field to a query

A calculated field is an added field in a query that displays the results of a calculation. For example, if we multiply together the hourly rate for each lesson and the length of each lesson, we can use the query to work out the cost of each lesson.

Type the name(s) of the field(s) to be calculated using the appropriate mathematical operators **(+, , *, / etc.**). For example if you wanted to multiply two fields together: [Field 1]*[Field 2]

2.11.1 Query 20 Calculating cost of lessons

7. Create a new query using Query Design and add **tblLesson**, **tblLessonType**, **tblInstuctor** and **tblStudent** 8. From **tblLesson** add **all** the **fields** to the grid

9. From t**blLessonType** add the **Cost** field to the grid.
10. From **tblInstructor** add the **firstname** and **Surname** to the grid
11. From **tblStudent** add **Firstname**, **Surname**, **Address1**, **Address2** to the grid

12. Rearrange the fields by dragging and dropping so that **InstructorID comes after LessonNO**, **followed** by **Instructor's FirstName and Surname**, followed by **Student's FirstName, Surname, Address 1 and Address 2**
13. Scroll to the **right** and find the **first blank column** of the QBE grid You are going to type the **calculation** in the Field row of the first blank column, like this:

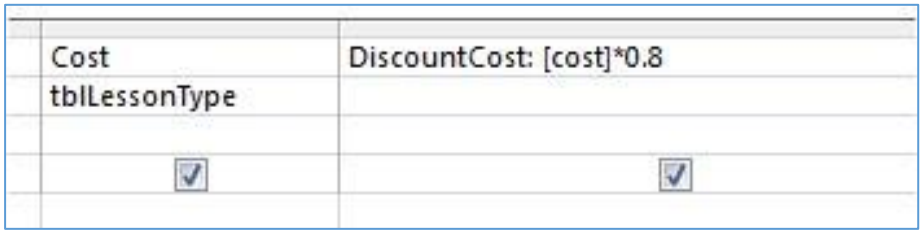
	tlkpLessonType	Cost	totalCost: [LengthOfLesson]*[Cost]
	tblLesson	tblLessonType	
	☑	☑	☑

In this example the query creates a new field called Total Cost and displays in it the value in the Cost field multiplied by the value in the Length of Lesson.

14. You are also going to add some **criteria** in the "**LengthOfLesson**" field that will select lessons that are greater than 1 (Look back at page 18 if you can't remember how to do this)
15. When you run the query you'll see all the students who have had lessons over an hour and the cost of that lesson.
16. Save this query as **qryLessonCost**

2.11.2 Query 21 calculating a discount cost

1. Open the query that you created in the last exercise **qryLessonCost** and select **File > Save As > Save Object As** and give it the name of **qryDiscountCost**
2. In the grid, go across to the far right column where you typed the calculation in. You are going to **edit** this to calculate a 20% discount.



In this example the query creates a new field called Discount Price and displays in it the value in the Unit Price field multiplied by 0.8. Note: This is the same as multiplying by 80% or subtracting 10% but the query does not recognise the % symbol and we have to devise an alternative expression.

3. Run the query and you'll see the discounted price for lessons over an hour long
4. Save the query as **qryDiscountCost**

3 Index

C

calculated field · 26 **criteria**
row · 12, 13, 15, 16, 23

E

Enforce Referential Integrity · 6, 7

L

Logical/Boolean operators · 15
Logical/Boolean Operators · 19

M

multi-table Queries · 22

P

Parameter queries · 20
parameters · 21

primary key · 6, 7

Q

Query Design · 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26

R

Referential Integrity · 6
relational operators · 15, 18 Relational operators · 18

relationship · 6

S

screenshot · 15

Select queries · 9

T

Totals row · 24

W

WildCard · 18