Lecture 4: More Microsoft Access® and Relational Databases

All screen shots from MS Access®
23 January 2002

Goal:

Introduction to the relational database model and intermediate queries (aggregation/grouping functions and multi-table queries).

Outline:

Discussion of databases, introduction to the relational model, and examples of the kinds of queries you might be asked to do in lab exercises.

Main Points:

Discuss data format and data structure
Basic Characteristic of relational database model
Intermediate queries
- aggregation function
- expression builder
- update database
- join multiple tables.

Databases Review:

Data format, data standards:

- Plain text: ASCII
- Formatted text: MSWord, WordPerfect
- Spreadsheet: *.xls, *wk3
- Database: dBase *.dbf

Difference between choice of data model (category of data class) and possible ways of standardizing it.

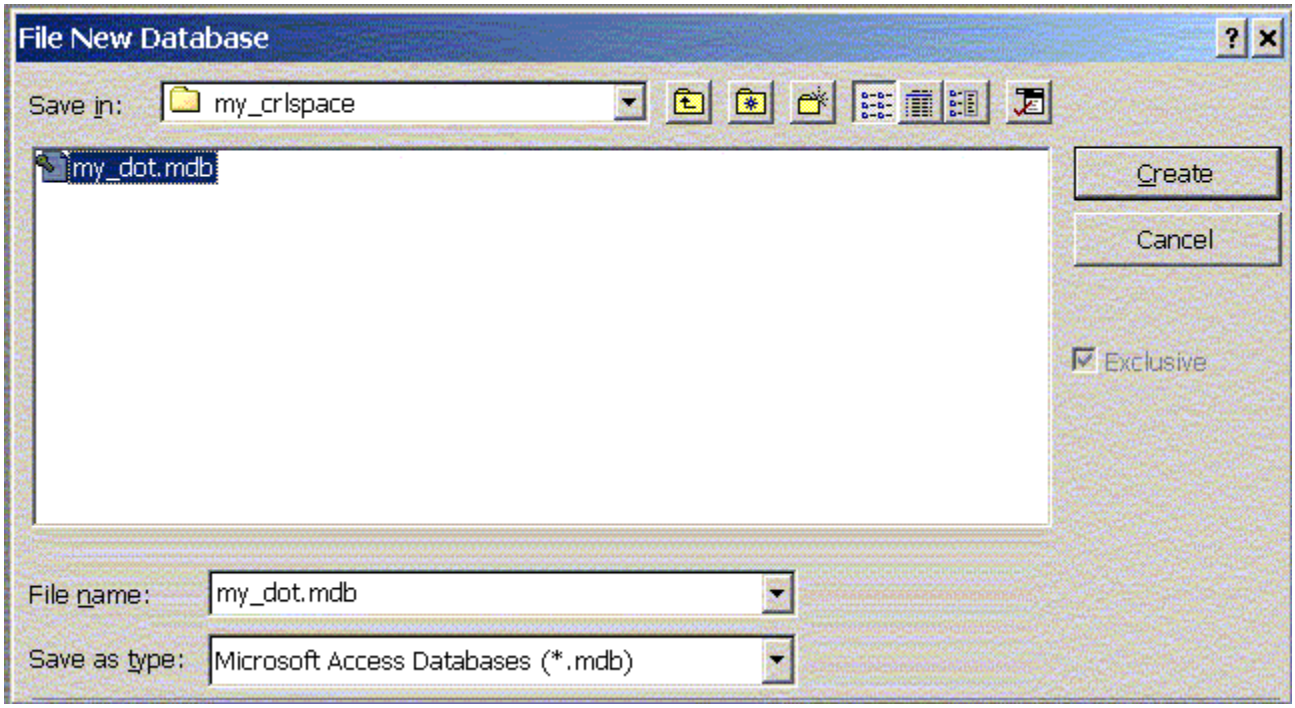
One way to see what a database is to see it in its ascii plain text format, common means of exchanging data among software programs. This format is also important because lots of information on the web is available in this most basic standard.

Data Structures and data models

- Logic structure to present real world situation
- How things of interest should be represented
- How things can be relatd to each other

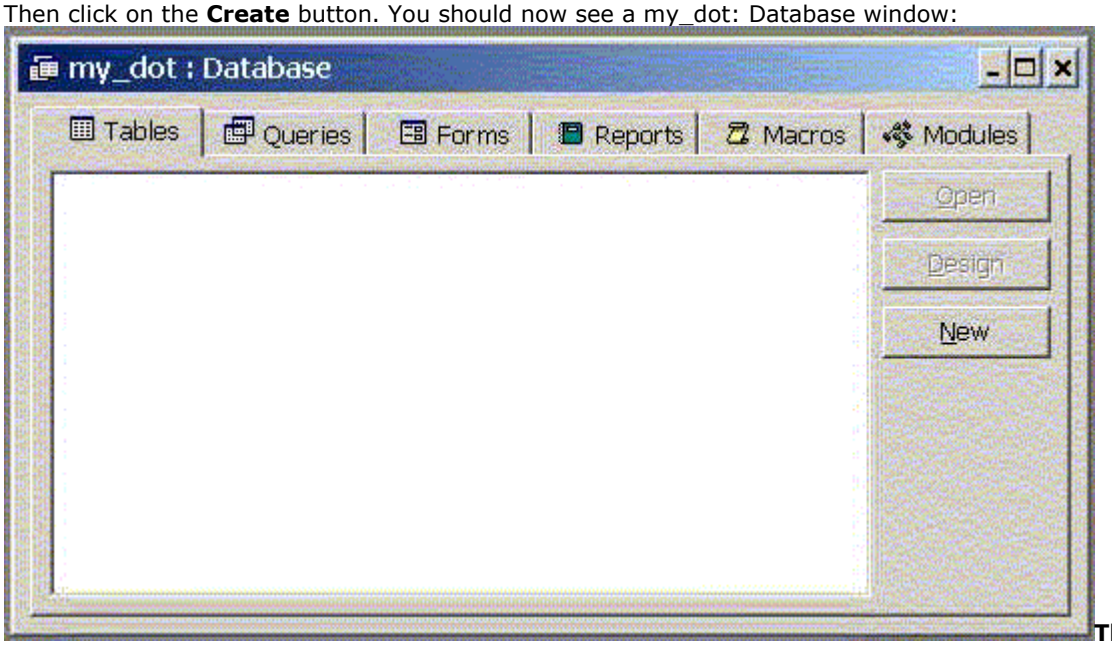
Simplest form of database is flat file (for relatively simple and small amounts of data)

Queries with flat file: Department of Transportation employee survey database.
Importing a type *.dbf file. To use a *.dbf file in MS Access, go to **File** then **New Database**... and with **Blank Database** highlighted, click **OK**. At the *File New Database* window, type in a File name such as **my_dot.mdb** and make sure that the **Save in** location is your crlspace private folder so that the window looks something like this:



Creating my_dot.mdb in *File New Database* Window

Then click on the **Create** button. You should now see a my_dot: Database window:



The new database window for my_dot

Now go to **File**... **Get External Data...** and then **Import**. In the Import window, you must specify where to look for the *.dbf file you want to import. In this case, go to K:\11.208\data\ and then be sure to specify Files of Type as dBase 4 (*.dbf). Now select **employee.dbf** and then click on the **Import** button. Access now displays a window saying "Successfully imported 'employee.'

Close the Import window. In the my_dot: Database window, you will see employee listed under tables.

Review of database documentation: data definitions and survey form.

Review basic query design

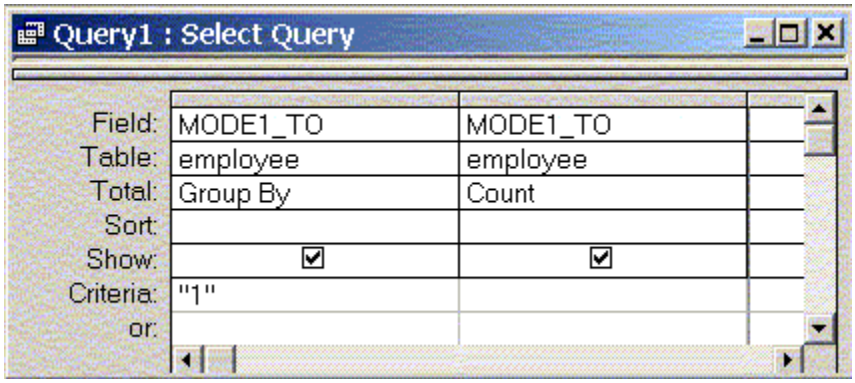
Use of **AND** versus **OR:**

- When you type expressions in more than one Criteria cell, Microsoft Access combines them using either the And or the Or operator. If the expressions are in different cells in the same row, Microsoft Access uses the And operator, which means only the records that meet the criteria in all the cells will be returned. If the expressions are in different rows of the design grid, Microsoft Access uses the Or operator, which means records that meet criteria in any of the cells will be returned.

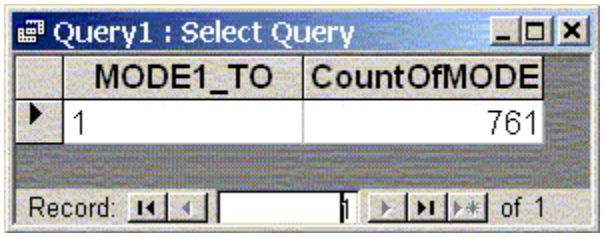
Aggregate functions (count, distinct); group by clauses

Expressions and calculated expressions (create new field first!)

- First a simple query: how many DOT employees drove alone as primary mode of transportation. First let's look at the data dictionary and survey.

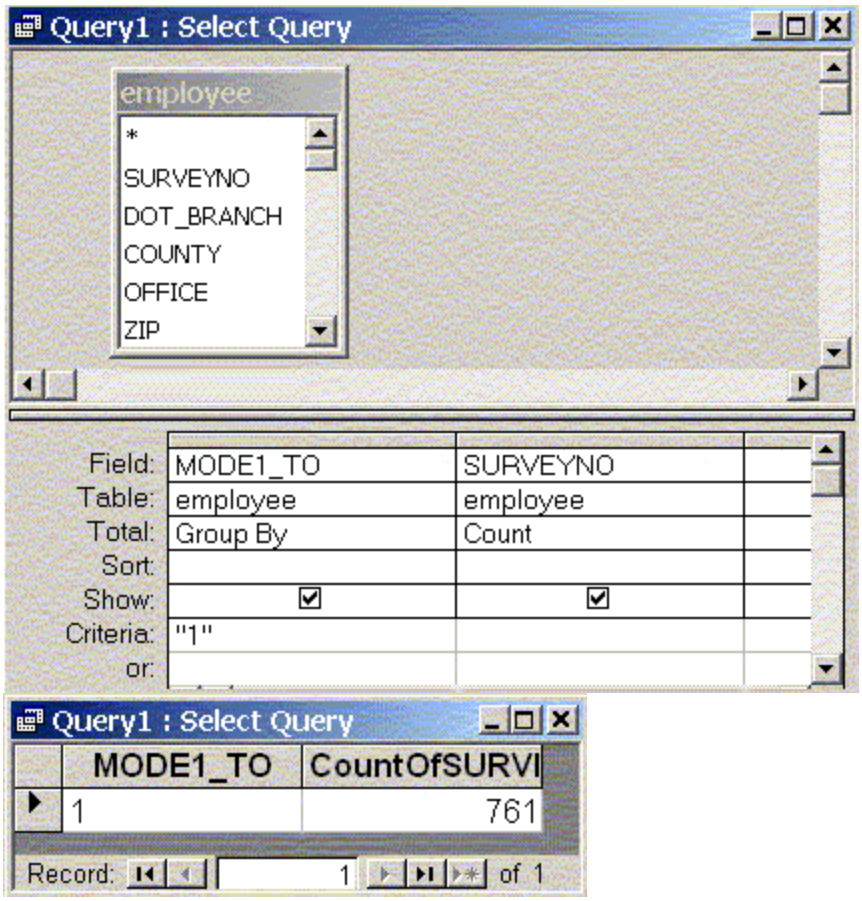


Query of employees who drove alone as primary mode

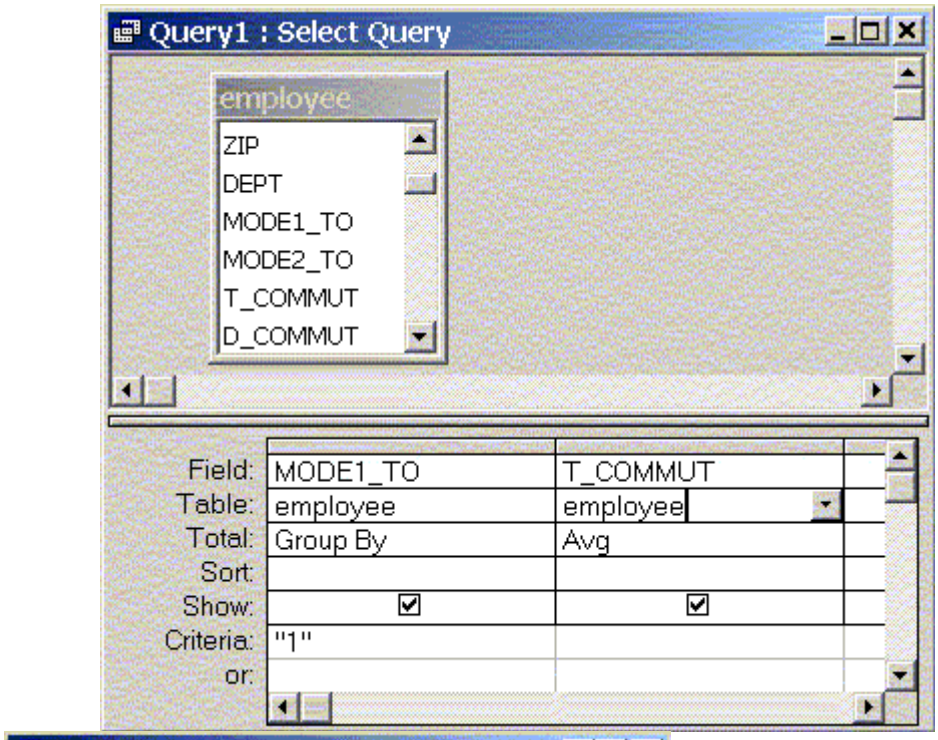


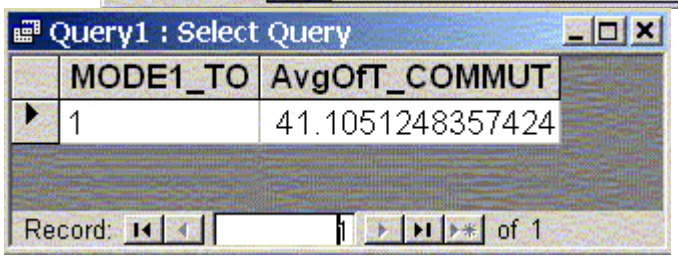
Result of above query

- Notice that we are looking at the record. So counting the unique survey identifier yields the same result.

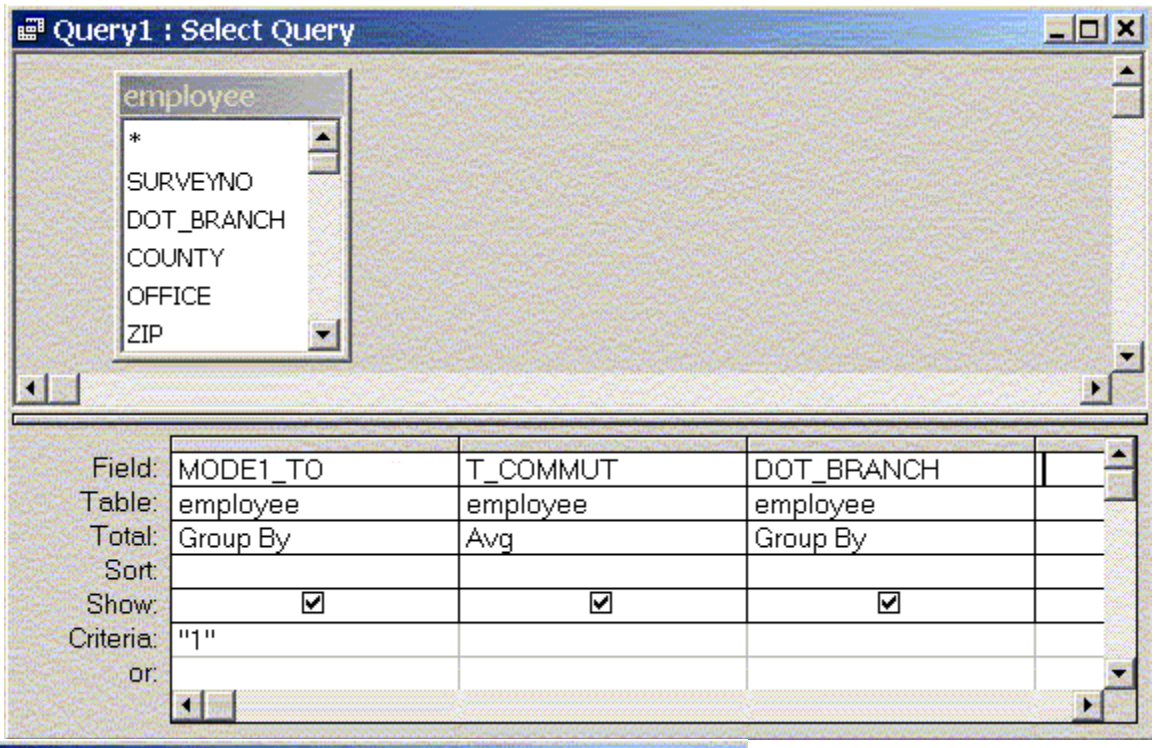


- Notice, however, that the Total functions can provide calculations on the selected field. For example, we can calculate the average commute time for those who drive alone.





- Group By functions also allow increasingly finer levels of selection. We can examine average commute times by DOT Branch...

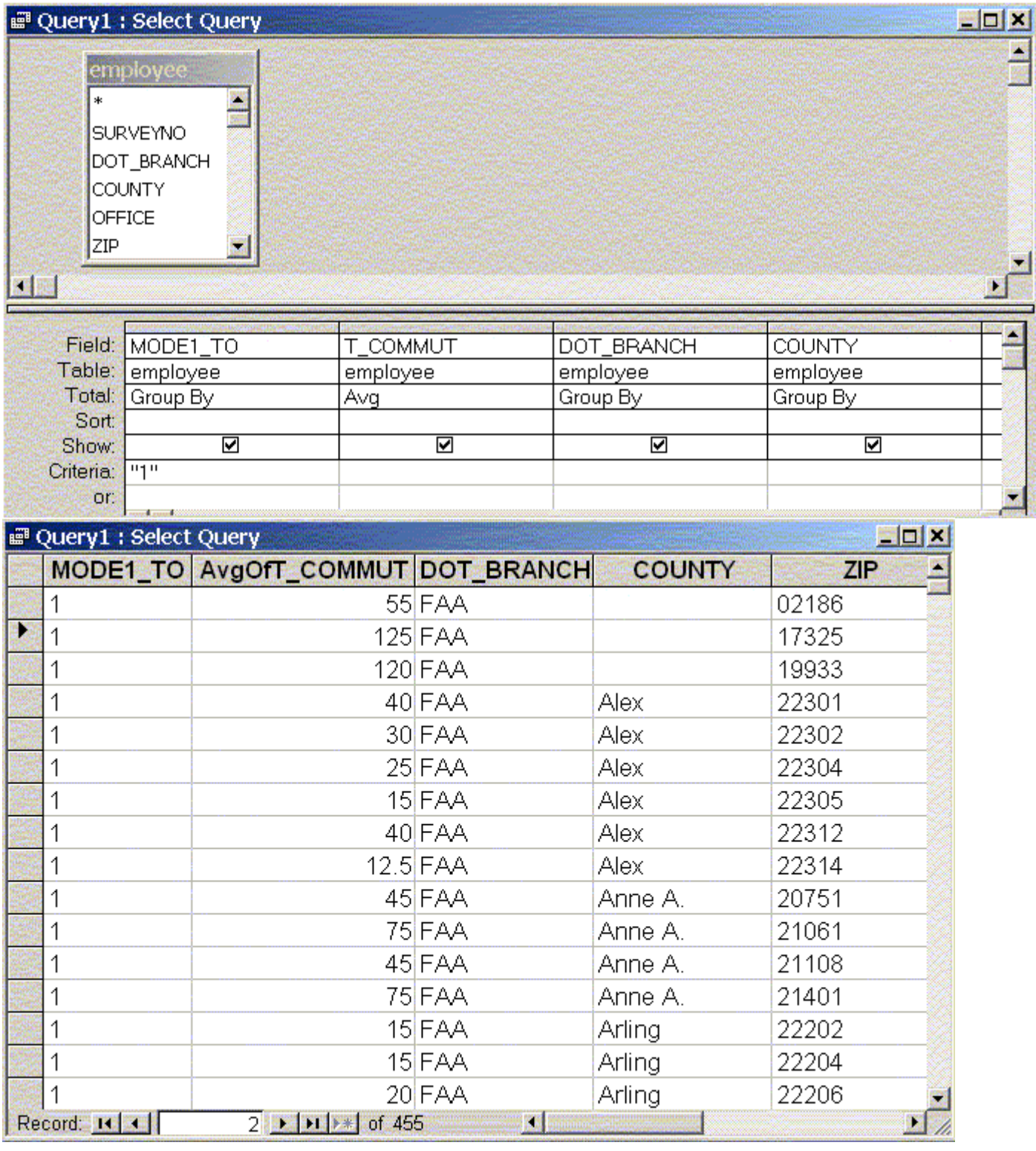


Query1 : Select Query

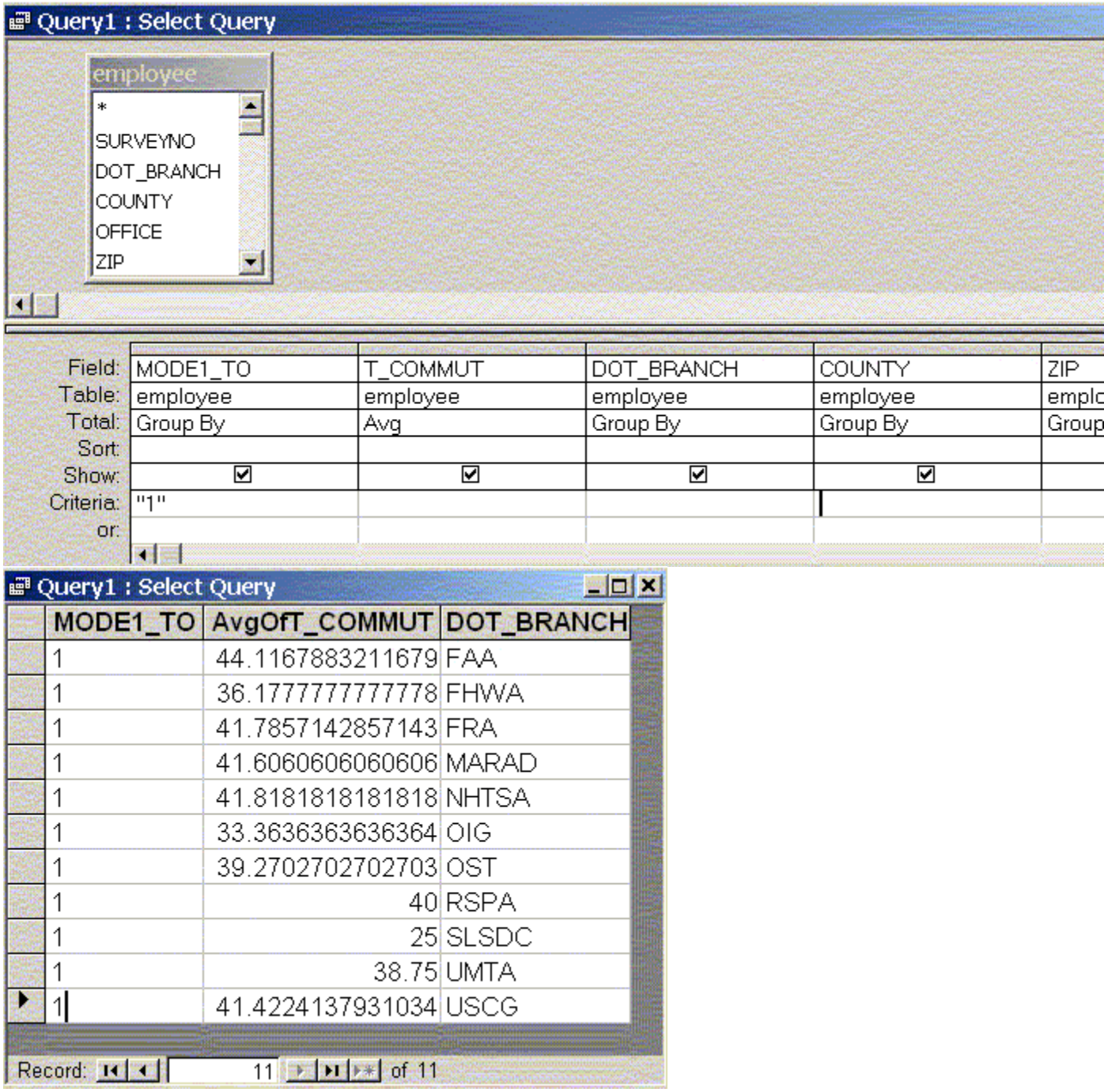
MODE1_TO	AvgOfT_COMMUT	DOT_BRANCH
1	44.1167883211679	FAA
1	36.1777777777778	FHWA
1	41.7857142857143	FRA
1	41.6060606060606	MARAD
1	41.8181818181818	NHTSA
1	33.3636363636364	OIG
1	39.2702702702703	OST
1	40	RSPA
1	25	SLSDC
1	38.75	UMTA
1	41.4224137931034	USCG

Record: 11 of 11

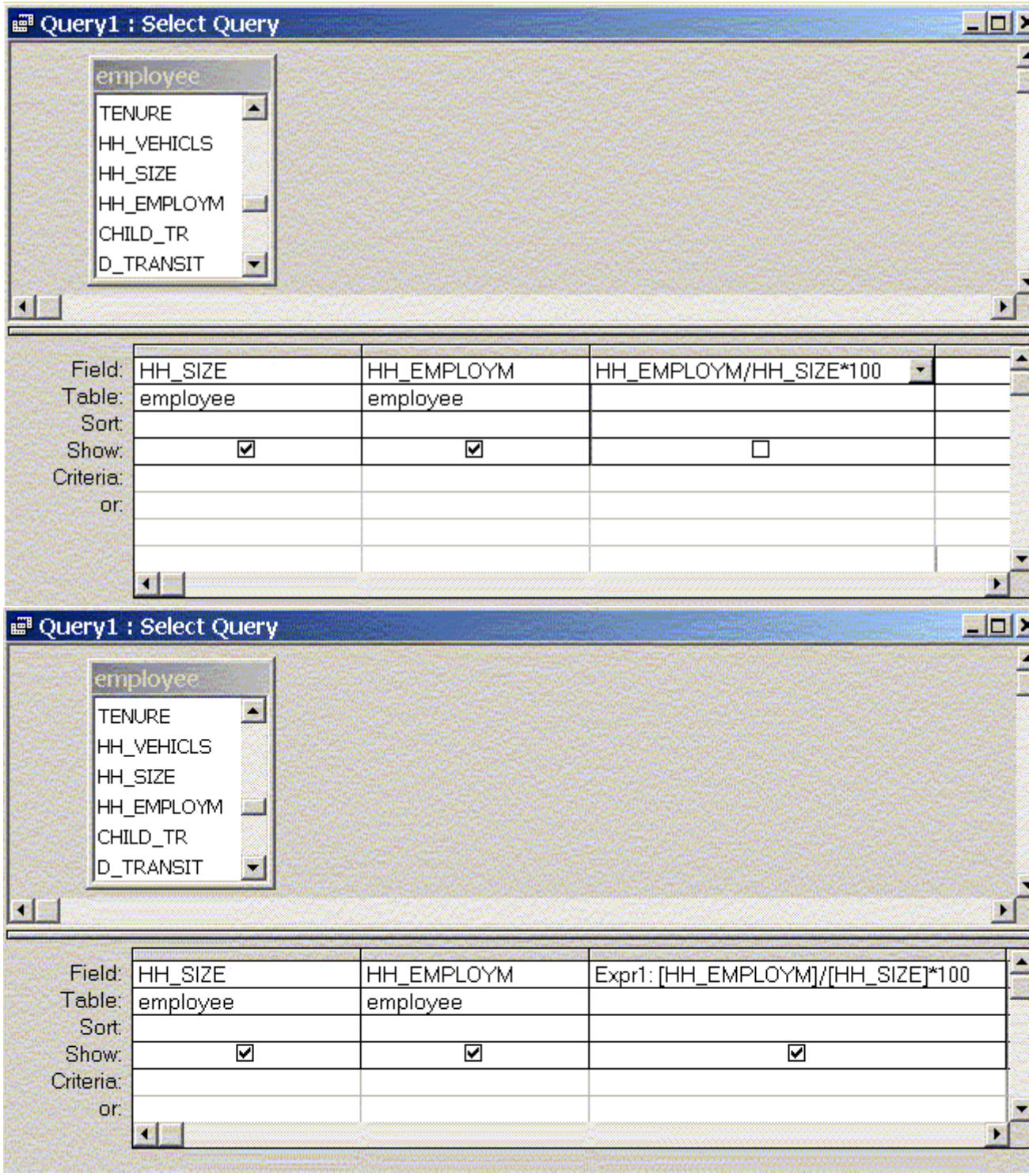
- And we can look further into the average commute times by county residence of particular DOT Branch employees...

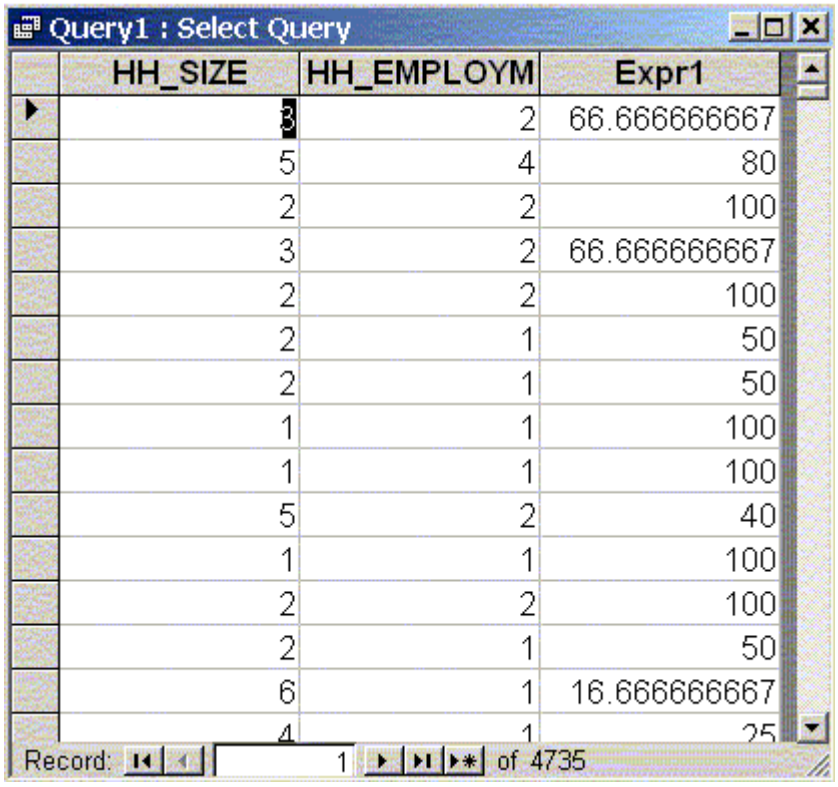


- And, of course, of Zip codes within counties....

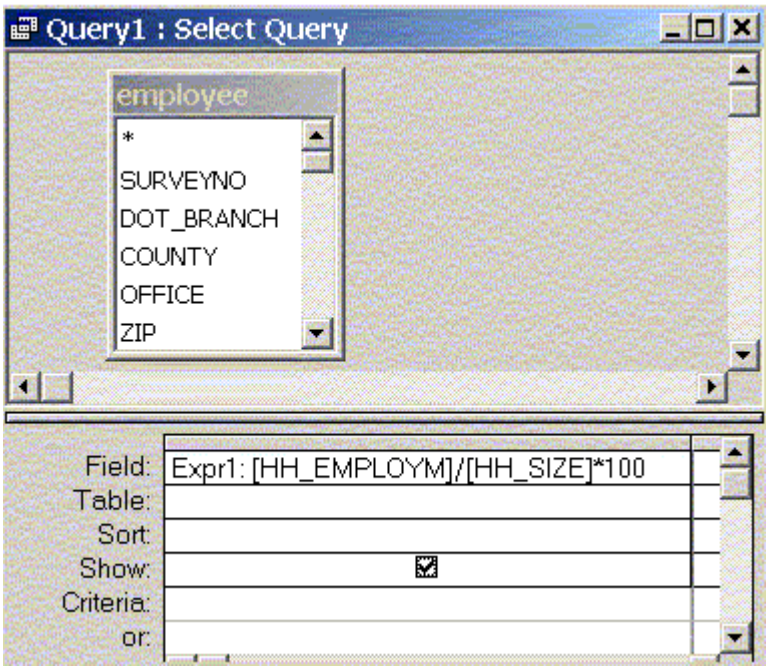


- It may also be useful to calculate expressions using two or more fields. We can number of persons in householod (HH_SIZE) and number of employed persons in household (HH_EMPLOYM). But what is percentage of persons employed?

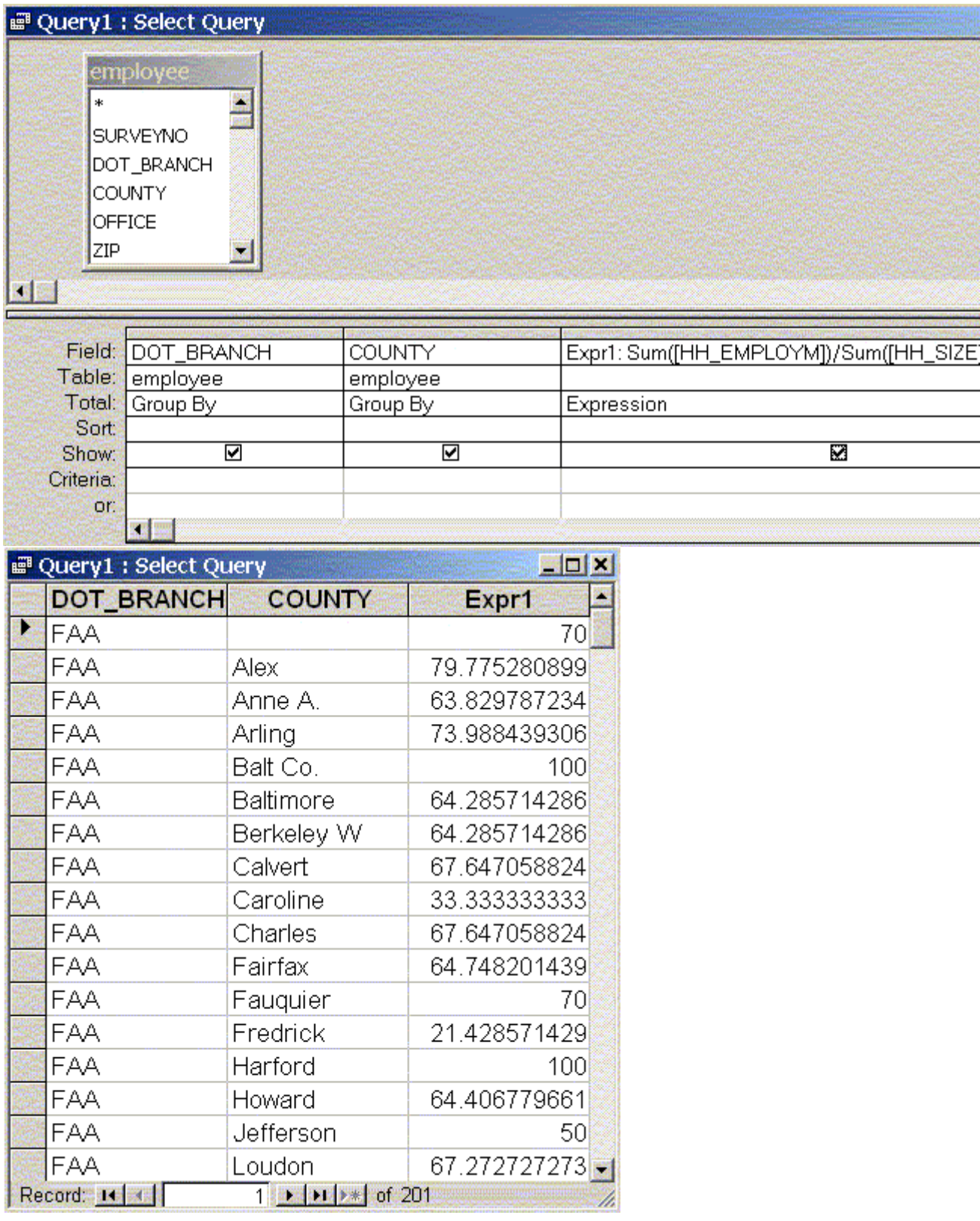

Query1 : Select Query
employee
TENURE
HH_VEHICLS
HH_SIZE
HH_EMPLOYM
CHILD_TR
D_TRANSIT
Field: HH_SIZE | HH_EMPLOYM | HH_EMPLOYM/HH_SIZE*100
Table: employee | employee
Sort:
Show: ☑ | ☑ | ☐
Criteria:
or:
Query1 : Select Query
employee
TENURE
HH_VEHICLS
HH_SIZE
HH_EMPLOYM
CHILD_TR
D_TRANSIT
Field: HH_SIZE | HH_EMPLOYM | Expr1: [HH_EMPLOYM]/[HH_SIZE]*100
Table: employee | employee
Sort:
Show: ☑ | ☑ | ☑
Criteria:
or:



- Of course, you only need the expression:



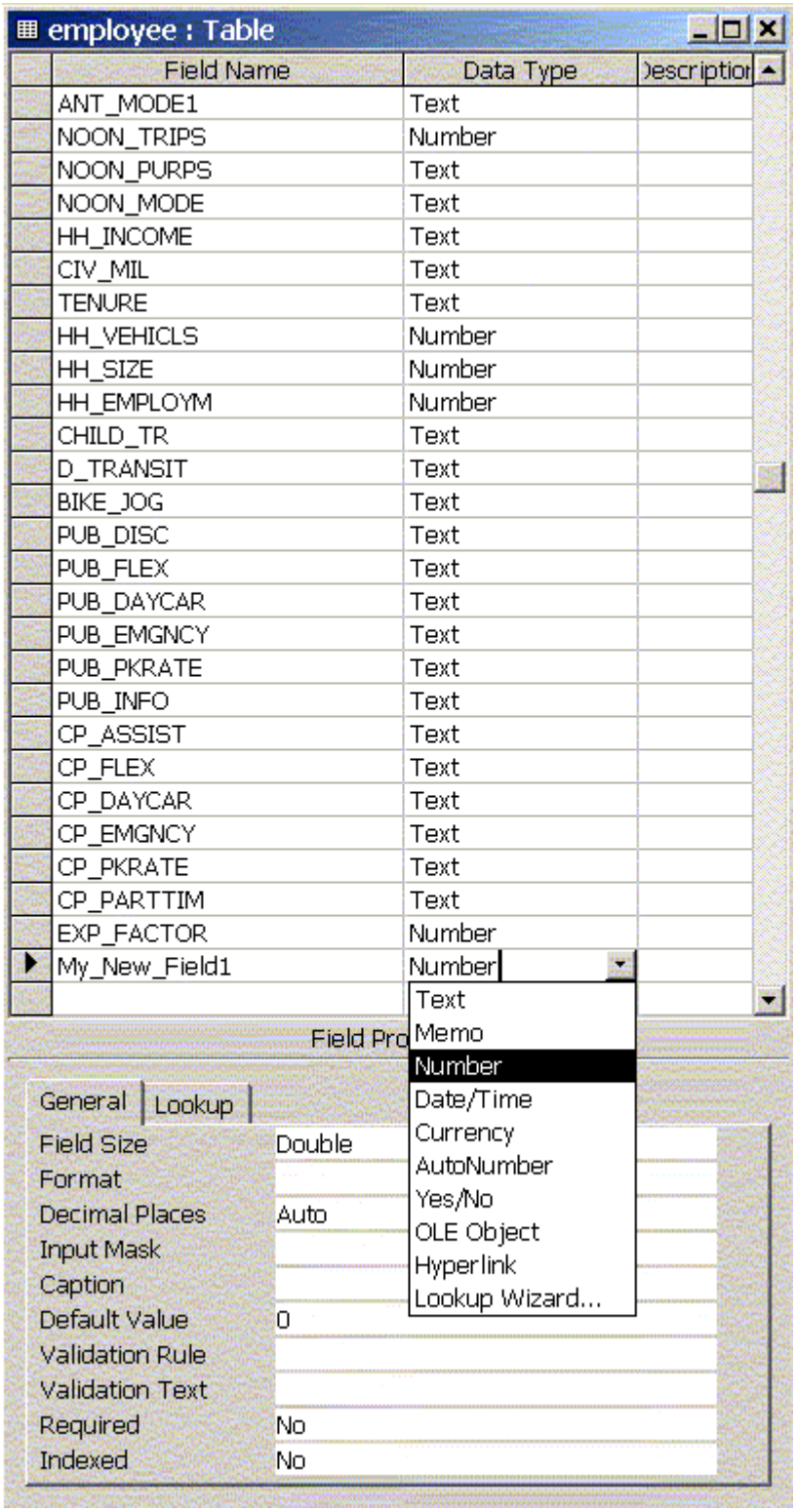
- However this does not allow us to use Group By functions. For that, we add Sums:



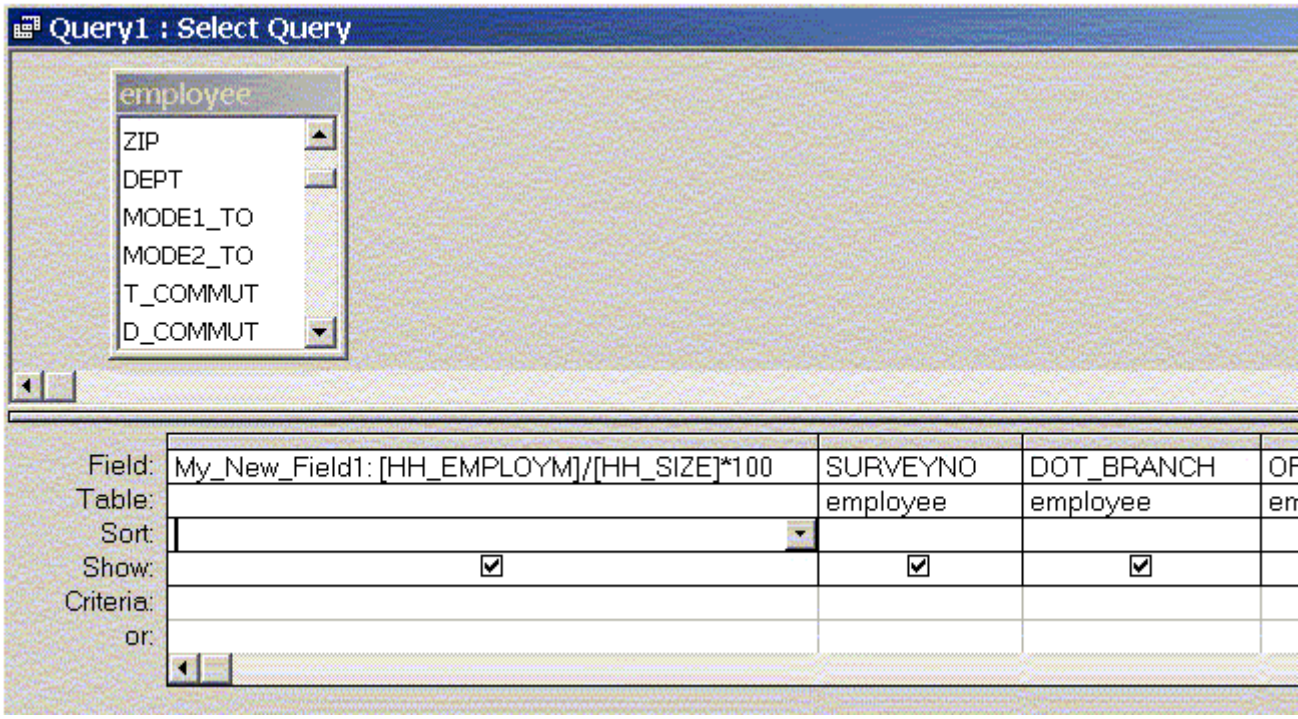
ADDING A FIELD AND CALCULATED EXPRESSION AND SAVING TO NEW TABLE:

- It also may be helpful to create a new field and then add data to it based on a calculated expression. At the main database window, go to **Tables** tab and then

click on **Design.** Scroll down the end of the employee table and click in the last empty row. Enter a field name (no spaces or odd characters). Then click inside Data Type cell, and a pull down menu appears. Select "Number." Also, in the gray area below under the "General" tab, select "Double." You window should like something like this:



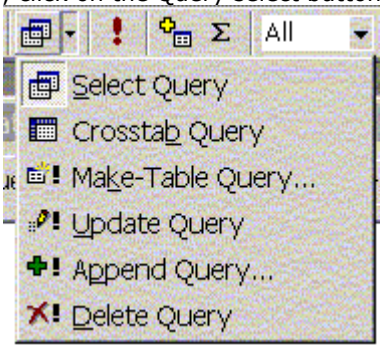
- Then close the table and save it. At this point, your table should now have a new column with no data in it.
- Let's create a new query with a calculated expression and field to include in new table:



Query1 : Select Query

My_New_Field	SURVEYNO	DOT_BRANCH	OFFICE	ZIP	
66.666666667	1	FAA	AAP	21045	28
80	2	FAA	APR	21043	28
100	3	FAA	AEE	21044	28
66.666666667	4	FAA	AAD	21044	28
100	5	FAA	AAP	21044	28
50	6	FAA	AFS	21044	28
50	7	FAA	ASC	21045	29
100	8	FAA	AEE	21045	28

Record: 8 of 4735

- At the Toolbar, click on the Query select button  and from the Query options menu 

select **Make-Table Query**. At the **Make Table** window, we enter a new table name and choose to leave it in the current database. After clicking on the Query now button  , Microsoft Access prompts us with a window to confirm that we will create a new table. We click **Yes**. We now have a new table that we can perform queries on.

Relational Databases and Joins

More complex model is relational database where there are multiple tables that can be joined by common identifier (just as social security number can be used).

Relational joins link different tables with different types of data. This generates information otherwise not possible. For example, you will join tables of building owners, parcels, and fires and determine which owner sustained the most property damage due to fire.

Example of relational joins using tables: parcels, fires, tax, owners

Create database "Parcels_plus_other"

Get External Data... and then **Import** the following table: **K:\11.208\Data\parcels.dbf** and then repeat for the tables **FIRES.DBF TAX.DBF** and **OWNERS.DBF**. Data Dictionary for PARCELS, FIRES, TAX, OWNERS

PARCELS (Parcel information)		FIRES (Fire incidents)		TAX (Tax and value information		OWNERS (Owner informatio	
id	unique parcel identifier	Id	Unique parcel identifier	id	unique parcel identifier	ownernum	owner identifier
wpb	ward/precinct/block	Wpb	Ward/precinct/block	wpb	ward/precinct/block	oname	owner name
parcel	parcel number in block	Parcel	Parcel number in block	parcel	parcel number in block	address	owner's street address
add1	street number	Fdate	Date of fire	prptype	property type	city	owner's city of residen
add2	street name	Ignfactor	Ignition factor	landval	value of land	state	owner's state of reside
zip	zip code	Estloss	Estimated loss	bldval	value of building	zip	owner's zip code of res
sqft	square footage			tax	property tax		
onum	owner identifier						
landuse	land use type						

Join tables with unique record identifiers: PARCELS.ID to FIRES.ID to TAX.ID. Then OWNERS.OWNERNUM to PARCELS.ONUM.

The same kinds of queries we ran on flat file we can run on joined tables. (Group by, Order by)

NOTE: Use of Criteria in joined tables:

- If your query includes linked tables, the values you specify in criteria on fields from the linked tables are *case-sensitive*. They must match the case of the values in the underlying table.

Relational joins can be helpful in other ways. Another example of adding a table and joining is to make analysis easier by providing lookup table to translate codes:

Open database myemployee.

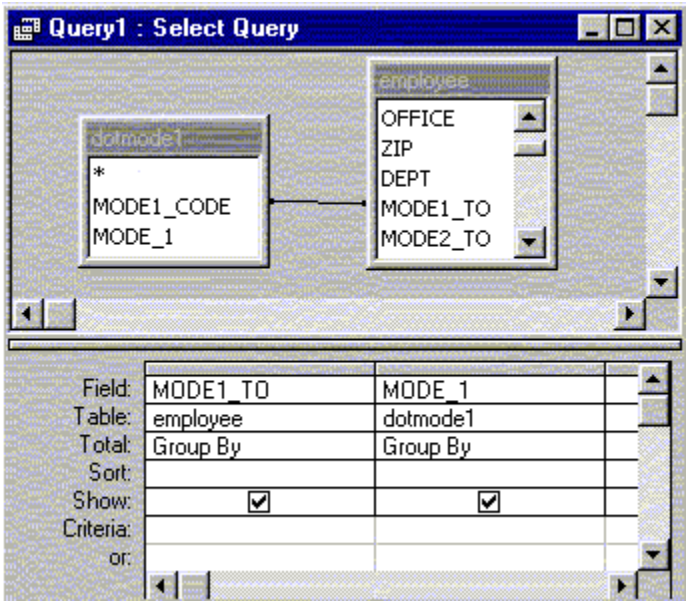
Use **Get External Data**... and then **Import** and in **K:\11.208\Data** select dotmode1.dbf and open to see it's a simple look up table:

dotmode1 : Table

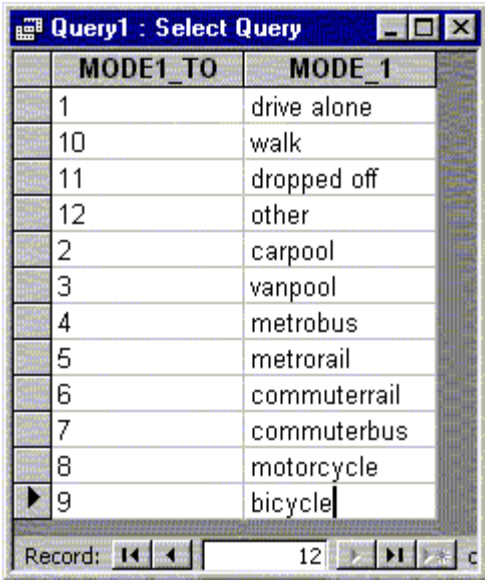
MODE1_CODE	MODE_1
1	drive alone
10	walk
11	dropped off
12	other
2	carpool
3	vanpool
4	metrobus
5	metrorail
6	commuterrail
7	commuterbus
8	motorcycle
9	bicycle
*	

Record: 7 of 12

Table contents of dotmode1



Employee and dotmode1 tables JOINED on MODE1_CODE to MODE1_TO



Result of above query