Exercise created by MaryJo Webster
@MaryJoWebster
Mjwebster71@gmail.com
July 2011 (updated March 2015)
Skills: group by queries, crosstabs, basic joining

Police chase data

Data is from the Minnesota Bureau of Criminal Apprehension. It contains data from between 1/1/2004 and 12/31/2010. It was used for a story in the St. Paul Pioneer Press in May 2011. While working with the data we discovered some flaws in the database, which led to us writing a sidebar about the problems. It was clear that this is a warehouse of information highly dependent on police agencies doing their part to submit the correct information — without anyone at the BCA vetting what comes in. But despite those flaws, it's still the most comprehensive source available about police chases in Minnesota.

Review the tables. Note the columns for the name of the police agency, date (a separate column for the year that PiPress added), and "injury type" and "collision" — which PiPress added. We created the field called "injury type" by running a query on the subject table that figured out which chases had 1 or more people injured or killed, then populated this new field accordingly depending on the findings from that query. We created the collision field with a "yes" if any of the following fields had a 1 (meaning "true"): propertydamagesquad, propertydamageviolator, propertydamageother.

Second table — called "Subject" — has one record for each person involved in the chase and indicates whether or not that person was injured and their role (violator, passenger, pedestrian, officer, etc)

We'll start by doing some basic queries just on the Pursuits table. We can get the majority of our answers just from this one table.

How many chases were there each year?

```
Select chaseyear,count(*)
From pursuits
Group by chaseyear
```

Notice that we don't have very many records for 2001 and 2003 and there are no records for 2002. The reason? This database just started going around that time and didn't really kick in until 2004. So we can really only use the 2004 to 2010 records for our analysis. (If you were doing this for a story, I'd recommend making a new table that only has the 2004 to 2010 records and working off that. We won't do that here, though)

Copy and paste these results into Excel and calculate the percentage change from 2004 to 2010. What's the trend?

Which county had the greatest number of chases across all the years?

```
Select county, count(*)
From pursuits
Group by county
Order by 2 desc
```

** remember putting "2" in the order by indicates that we want to sort by the second column in our answer (in this case, the count of chases) and that we want it to go in descending order (desc)

Which police agency had the most chases across all years?

```
Select name, count(*)
From pursuits
Group by name
Order by 2 desc
```

Which county had the greatest number of chases in 2010?

```
Select county, count(*)
From pursuits
Where chaseyear=”2010”
Group by county
```

**adding the Where clause to the same query simply limits which records will be used in come up with our answer

What’s the most common reason for a pursuit?

```
Select reasonforpursuit, count(*)
From pursuits
Group by reasonforpursuit
Order by 2 desc
```

What’s the average number of miles a pursuit goes?

```
Select avg(miles)
From pursuits
```

Does that differ from year to year?

```
Select chaseyear, avg(miles)
From pursuits
Group by chaseyear
```

How many of the pursuits resulted in an injury or death?

*Remember we have the “injurytype” field in the pursuits table that indicates whether one or more people was injured or killed (if it says “no injury” it means there was only property damage). So we can get the answer to this question from the pursuits table. If we tried to get this answer from the subject table, our answer would be the “number of people”. For this question we want the “number of pursuits”

There are a couple different ways you could do this query. If you just want a single number, you could do this:

```
Select count(*)
From pursuits
Where injurytype=”injury” or injurytype=”fatal”
```

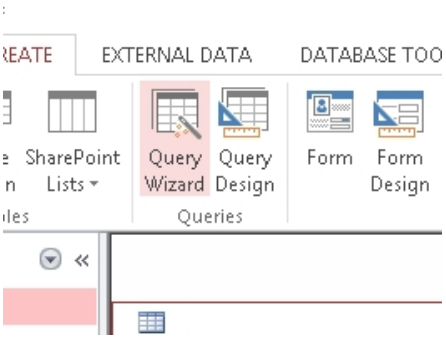
If you want to try to put the injuries/fatalities into context with all the chases, then you could do this:

```
Select injurytype, count(*)
From pursuits
Group by injurytype
```

Copy and paste this into Excel and calculate percent of Total (will need to add together Injury and Fatal and then divide by the grand total)

Next, let's figure out if there's any variation among the police departments in terms of percentage of chases that result in injury or death.

We're going to run a crosstab query – this type using the "Query Wizard"

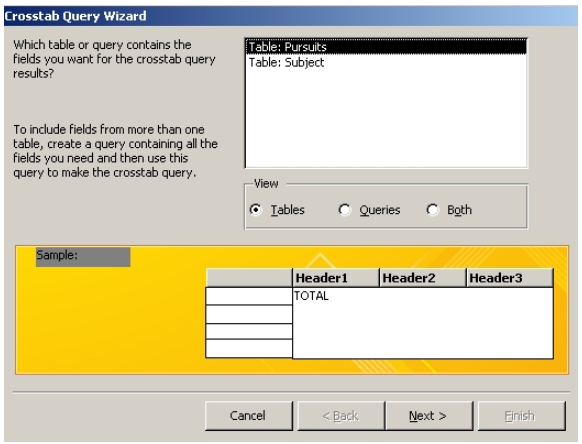


When you click on Query Wizard, it will first ask what type of query you want to run. Choose the second option, "Crosstab query wizard"

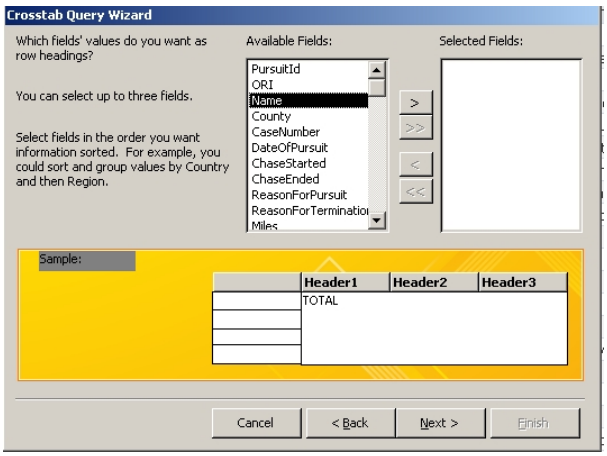
Click OK

Next, choose the table you want to use for your query – in this case, we want "Pursuits"

Click Next.



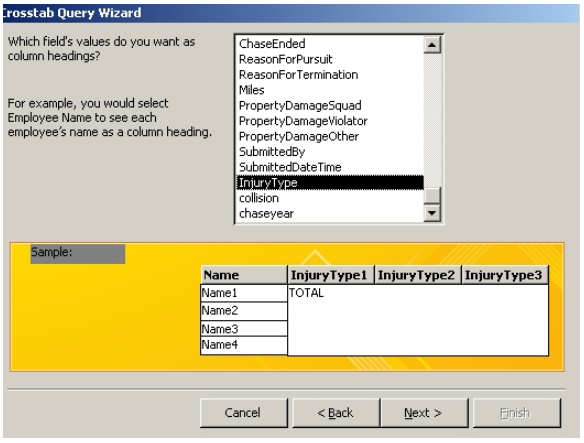
Next, it's asking what field(s) we want as the "row headings." What this is saying, is what do you want your rows to be? In this case, we want to list the names of the police agencies – stored in the field called "name"



Click on "Name" in the left window and push the arrow button > to push it over to the "selected fields" box on the right side.

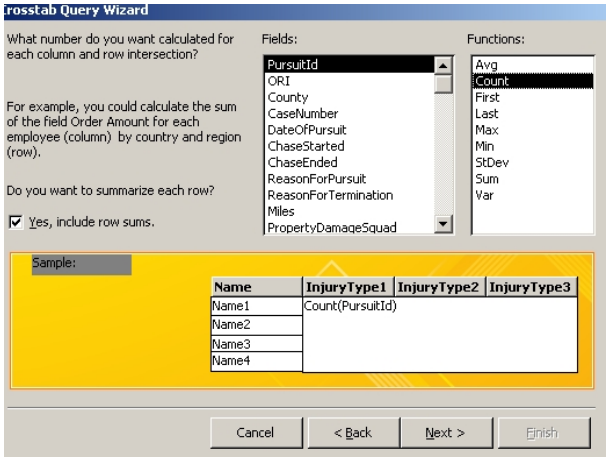
Click Next.

Next, you need to tell it what you want as the columns. In this case, we want "Injury Type"



Click Next

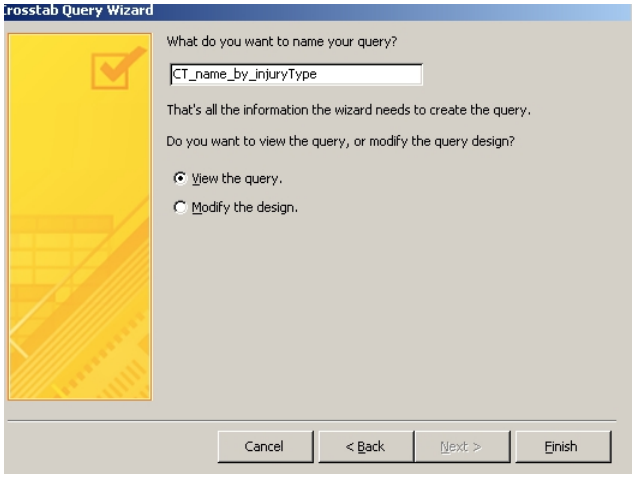
Next, we want to get some data. We need to tell it to count up the number of chases. You can choose anything from the list of fields (I'm going to just leave it on the first field) and then choose "Count" from the functions list.



Click Next

Finally, we can name the query (it's going to save it automatically). I'm going to start mine with "CT" for crosstab and then list the fields that it's crosstabbing – "CT_name_by_InjuryType"

Click finish



Your results should look like this. Note that it put in a total pursuits field for us (total of pursuitID" and then it tallies up the pursuits by injury type. You can copy and paste this out to Excel to calculate percentages. I would recommend first populating the blank spaces with zeros (you can do a "Replace All" just like you would in a Word document). Add together the "fatal" and "injury" columns and divide by the total to get the percentage.

CT_name_

Name	Total Of PursuitId	fatal	Injury	No injury	Unknown
ADA POLICE DEPARTMENT	1				1
ADRIAN POLICE DEPARTMENT	1	1			
AIRPORT POLICE DEPARTMENT	3		1	1	1
AITKIN COUNTY SHERIFF OFFICE	7		1	4	2
AITKIN POLICE DEPARTMENT	1			1	
ALBERT LEA POLICE DEPARTMENT	21		5	16	
ALEXANDRIA POLICE DEPARTMENT	15	1	5	6	3
AMBOY POLICE DEPARTMENT	2			2	
ANOKA COUNTY SHERIFF OFFICE	44		12	28	4
ANOKA POLICE DEPARTMENT	11			11	

We saw in our first query that the number of chases has come down significantly in later years. Were there any police agencies in particular that had big drops in the number of chases?

The easiest way to get this answer will be to create another crosstab. This time put "Name" as the rows (like we did in the last one) and put "chaseyear" as the columns. Again, count the number of chases.

Then you can export the results to Excel and calculate the percentage chase for each one. Sort the results, to see who had the biggest drop.

What day(s) of the week are chases more likely to take place?

Our data doesn't have a field indicating if it was Sunday, Monday, Tuesday, etc., however we do have a date of the pursuit. Access (and most other data software programs) have all kinds of great functions for converting dates to something more useful. Let's put a couple of them to work.

WEEKDAY() will convert the date to a number that represents the day of the week. 1 is Sunday, 2 is Monday, etc.

MONTH() pulls out just the month
DAY() pulls out just the day
YEAR() pulls out just the year

So to count up the chases by the day of the week that they occurred…..

```
SELECT weekday(DateofPursuit), count(*)
From pursuits
Group by weekday(DateofPursuit)
```

We could also see if there are any patterns by month of the year…..

```
Select month(DateofPursuit), count(*)
From pursuits
Group by month(DateofPursuit)
```

What percentage of chases result in a collision?

Remember we have a field that I added indicating if there is a collision or not – either a yes or no.

```
Select collision, count(*)
From pursuits
Group by collision
```

*Paste the result into Excel to calculate the percent of total

Now we'll use the Subject table to get some more information and use some more advanced queries.

Let's start with just the Subject table, to gather some basic information.

What's the average age of the violators?

First, let's check the integrity of our data. I see that there are a lot of blank spots for the age field in the Subject table.

If we want to focus on violators, we need to find out what percentage of the violator records have an age filled in.

First, we need to know how many total violators there are.

```
Select *
From subject
Where subjecttype= “violator”
```

Answer=6896

Then, we need to know how many of those records have a blank spot for the age

```
Select *
From subject
Where subjecttype= “violator” and age is null
```

Answer=1105

So that means 16% of the violator records don’t have an age filled out. That percentage is a bit high, but not too far out of whack that we couldn’t use it. Less than 20% is usually OK — but I’d prefer less than 10%.

Now we can calculate the average age of violators:

```
Select  avg(age)
From subject
Where subjecttype= "violator"
```

Answer: 29

How many “bystanders” were involved in pursuits?

The field called “subjecttype” indicates who the record refers to. Let’s see what’s in this field. Run this query:

```
Select subjecttype, count(*)
From subject
Group by subjecttype
```

Query2	
subjecttype	**Expr1001**
Officer	2952
Party in Unrelated Vehicle	164
Unrelated Pedestrian	13
Violator	6896
Violator Passengers	2397

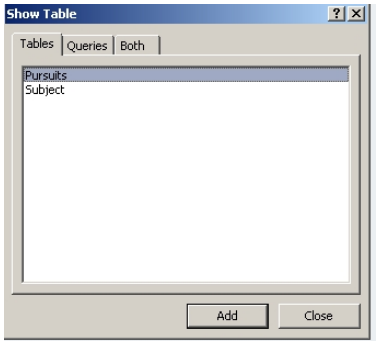
You’ll see that we have 2 values that would indicate the person is a bystander and in both cases, they have the word “unrelated” in the field. That will make it a little easier for us to write the queries to focus on them in our next query.

The two tables in this database were designed to be joined together. This is a one-to-many relationship — there is one record for each chase in the Pursuits table and there could potentially be many records for each person in the Subject table. (it’s also possible there won’t be any records in the Subjects table)

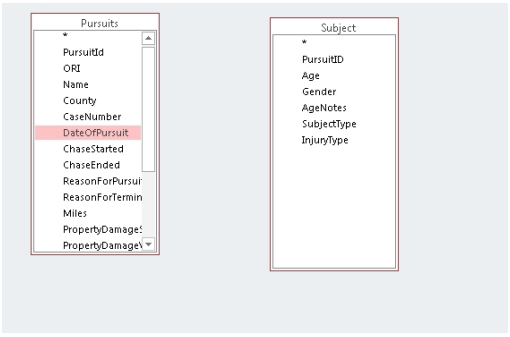
Let's find out how <u>frequently</u> "bystanders" are involved in pursuits. This is a different question than the previous one. The previous one was counting people. This time we want to count chases.

We need to start by isolating the chases that have a bystander involved. So we will need to join the tables together.

Start a new query and when the "Show Table" box comes up, put both tables into the query.

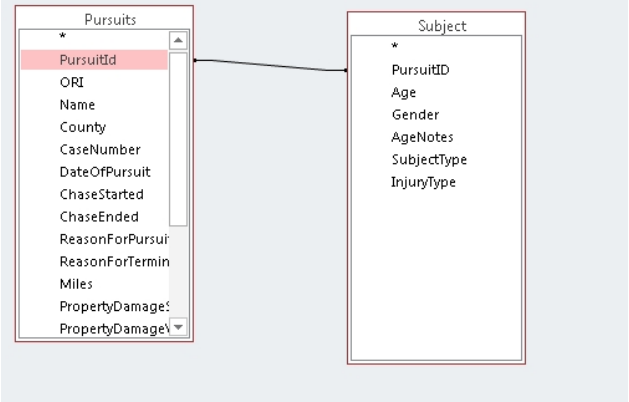


Stay in the Query Design window and you'll see the two tables – listing out their fields – like this:



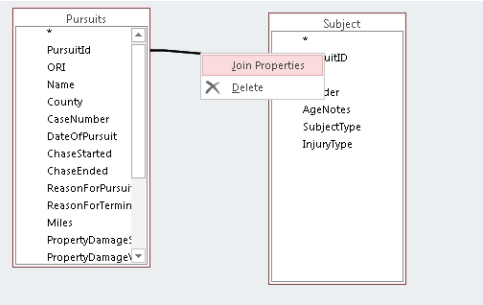
To join them, click on PursuitID in Pursuits, hold down the button on your mouse, and drag across to PursuitID in the subject table. Let go of your mouse when it's hovering over the PursuitID field in the subject table.

You'll end up with a line like this.



Right-mouse click on the line and bring up “Join Properties”

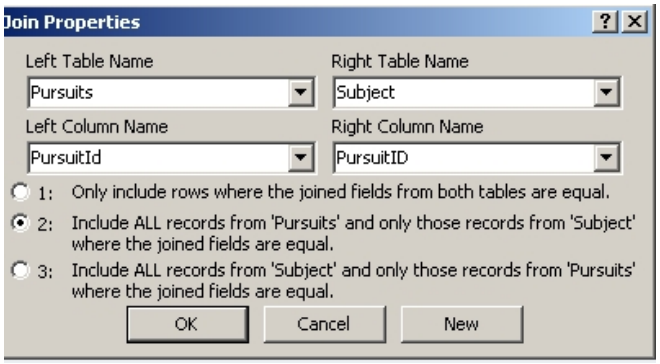
This will enable us to make sure the join worked.



Here you can see the table names and field names that are matching. If you made a mistake on the fields, you can correct it here.

Then you’ll see the 3 radio buttons. This is how you designate the type of join.

The first option is the default. It will return only the records where there are matching records in both tables. So for example, if there is a pursuit listed in pursuits but there are no records for that pursuit in the subject table, then that pursuit will NOT be included in our answer. It could work the other way too, that there could be records in the subject table, but no matching pursuit record in the pursuits table (this is known as “orphan records.” It’s something you want to look for when joining tables)



In this case, it’s legitimately possible that there could be a pursuit record, but no subject records – so it’s best if we choose the second option to ensure that all the chases are included in our results.

Now we can build our query. Choose the following fields from pursuits: PursuitID, Name, County, DateofPursuit, Collision, ChaseYear. And choose the following fields from Subject: SubjectType, InjuryType

Flip over to the SQL View and you should see this:

```
SELECT Pursuits.PursuitId, Pursuits.Name, Pursuits.County, Pursuits.DateOfPursuit, Pursuits.collision, Pursuits.chaseyear, Subject.SubjectType, Subject.InjuryType
FROM Pursuits LEFT JOIN Subject ON Pursuits.PursuitId = Subject.PursuitID;
```

You’ll see that the join syntax is in the FROM line.

Go ahead and run this query. You’ll get 12,445 records. Let’s look at what this did. Remember that we have 12,422 records in the subject table (in other words, that’s how many people we have).

When you join two tables that have a 1 to many relationship (in this case, 1 pursuit to multiple records for the people), your result is going to be based on that “many” table.

But why didn’t we get 12,422 records?

Remember in our join we chose the 2nd option – return all the records from Pursuits, and only those that match from Subject.

If you filter your query results, you'll see there are records where the "subjecttype" and "injurytype" fields are blank. These are records where there is a pursuit record, but no matching records in the subject table.

So we have 23 pursuits without people and 12,422 people = 12,445 records in our answer.
We need to adjust this query a little bit to get back to our original question – how many pursuits involved bystanders?

Right now, our answer has everybody. We need to limit it to just the bystanders. In this case, any records where the subjecttype includes the word "unrelated."

Go back to the SQL view of your query and add this on the end (don't forget to remove the semi-colon!):

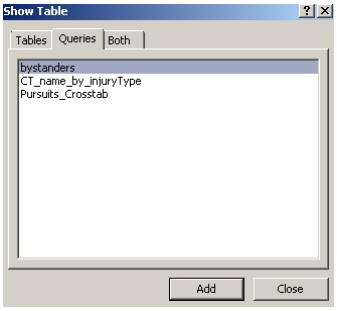
where subjecttype like "*unrelated*"

Then run the query. You should have 177 records. That matches what we got in our last query – if you add together the two "unrelated" fields.

But this still doesn't tell us how many "chases" involved 1 or more bystanders because there are some chases that had multiple bystanders. So we need to winnow this down to 1 record for each chase.

SAVE this query. Name it "bystanders" and close it.

Then launch a new query, this time based on the "bystanders" query.



We're going to use a function called DISTINCT that will only return 1 record.
Look at "bystanders" again and fine PursuitID # 436. You'll see there are 2 records. The fields that we pulled from the pursuit table (pursuitID, name, county, dateofpursuit, collision and chaseyear) are exactly the same for both records. Only the fields we got from the subject table (subjecttype, injurytype) are different.

If you add DISTINCT to a query that has multiple records with the same information, it will only return 1 of each one.

So if we run this query, it will give us just one record for each of the chases in this "bystanders" list:

SELECT DISTINCT pursuitid, name, county, dateofpursuit, collision, chaseyear
From bystanders

You should get 112 records. And remember we had a total of 6,899 chases. That means bystanders are involved in only about 2% of all chases.

You could refine this question a bit and ask "what percentage of chases result in an injury or death of a bystander?"

To do that, filter your “bystanders” query so that you also limit it to only those where “injurytype” <>”No injury”

```
SELECT Pursuits.PursuitId, Pursuits.Name, Pursuits.County, Pursuits.DateOfPursuit,
Pursuits.collision, Pursuits.chaseyear, Subject.SubjectType, Subject.InjuryType
FROM Pursuits LEFT JOIN Subject ON Pursuits.PursuitId = Subject.PursuitID
where subjecttype like "*unrelated*" and subject.injurytype<>"no injury"
```

Save the query again. And then re-run the DISTINCT query.

You end up with 60 chases – less than 1%