

Joining Tables and working with federal campaign finance data

Access database called FEC0809 (also need 3 dictionaries from FEC)

This database contains three tables from the FEC:

ContMN: contribution records for all donations in the 2007-2008 election cycle that came from donors listing MN as their state OR went to committees that were based in Minnesota.

Comm: All committees registered with the FEC for that election cycle.

Cand: All candidates registered with the FEC for that election cycle.

NOTE: the individual contributions table shows the recipient ID (Recip_ID). This ALWAYS matches to a committee in the committee table. If you want to find out what candidate got money, you'll have to join committee to candidate (Use all three tables in your query) to get the candidate names. Typically the candidate's committee name includes their own name, however there are some exceptions, and it's often likely that a candidate will have more than one committee.

The data dictionaries for each table — provided by the FEC — are included.

This data was downloaded from: <http://www.fec.gov/finance/disclosure/ftpdet.shtml>

Let's start by getting to know our data a little bit.

Which Minnesotan donated the most money in the 2007-08 election cycle? (regardless of who they gave it to). Note: we need to make sure we don't lump together people who have the same name.

This is one situation where you want to think carefully about what field(s) be used to designate your groups. In this case, if we simply used CONTNAME, we could run into problems. Why? Because of people with common names like "Bob Smith." So we need to find a way to separate the Bob Smiths. One way would be to include the city. A more precise way would be to include the Zip code if you have it (which we do). Another option is the occupation field — but you would have to make sure that it's standardized. It's quite common in campaign finance data to have one donation from Bob Smith say "retired" and another to say "retired; former military" or something like that.

The bottom line is that anytime you want to do a query like this — determine who gave the most, or who got the most or which group gave the most, etc — you will likely run this query initially to get a sense for what you're going to find, then you'll have to spend some time doing some data cleanup to get the answers that you will actually publish.

```
Select contname, zip, sum(amount)
from contMN
Where state= "MN"
group by contname, zip
order by 3 desc
```

Some interesting names pop up at the top.

First let's take a look at Michael Ciresi. Do a query on the candidates table and see what comes up under that name.

```
Select *
from cand
where name like "ciresi*"
```

Yep, he's running for office. So we're going to have to see if all that money went to his own campaign or not. (Save this for a bit later)

The second and third names in our contributor list — Al and Jayne Hilde — are a Minneapolis couple with an interesting story. Al built the first prototype of a "mobile toilet cabin" — more commonly referred to as the port-a-potty! He created a company called Satellite Industries in 1958, and sold the company to his son in 2004. Now they give a lot of money to charities, such as the Minnesota Orchestra.

Then a couple more names down the list and you start to see quite a few Pohlads. These are likely members of the Pohlad family that own the Minnesota Twins baseball team.

The big thing missing from those queries is who the donations were made to. The ContMN table doesn't tell us the name of the committee that got the money — it just has an ID number. That

RECIP_ID number matches with a field in the COMM table, where it does display the committee name and other details about the committee.

So let's start with a basic join.

We'll join the ContMN and the COMM table and we'll return all records and display a few fields from each table. You can create the join several ways — the first two in SQL and the third using the Grid View.

1) In the FROM line, using an INNER JOIN (this type of join only returns records where it finds a match between the two tables. There are other joins that are more advanced that would allow you to get all records from one table, plus whatever records match from the other table)

```
SELECT CONTNAME, CITY, CONT_DATE, AMOUNT, NAME
FROM comm INNER JOIN CONTMN ON comm.ID = CONTMN.RECIP_ID
order by amount desc
```

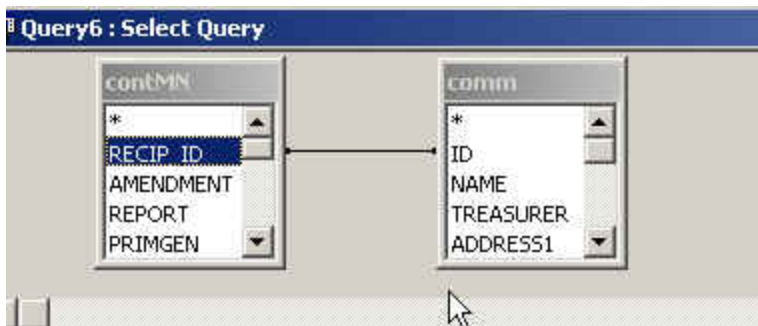
2) In the WHERE line of your SQL. This seems easy, but it gets more complicated when you want to add filters to your data (i.e. where cand.name= "coleman, norm")

```
select contname, city, cont_date, amount, name
from comm, contmn
where comm.id=contmn.recip_id
```

3) In the grid view, you can very easily make the join when you're creating your query in the first place. Create a query, put both tables in the query. Then in the design window click on the RECIP_id field from the ContMN table, hold down your mouse, and drag over and let go of the mouse when your cursor is hovering over the ID field in the Comm table. When finished, it will have drawn a line like this:

Note: you can get rid of the join by right-clicking on the line and choosing "Delete"

Once we have the two tables joined, we can choose fields from either of the two tables to



display. Just remember that when you join two tables using an INNER JOIN (which is the default join), it's only going to return records that have a mate in the other table. So for example in this case, if there happened to be one record in the ContMN table where the recipient ID number was messed up, that record wouldn't show up in our results. This is something known as "orphan records."

We can tell if the join got everything by looking at the record count on the query we just ran. Our ContMN table has 65,587 records. And the query shows the same number. Great! Our join works perfectly. Every contribution record found a mate in the committee table.

Let's go back to Michael Ciresi and see who he donated his money to. Was it his own campaign? Start by running a simple query that shows each of the donations:

```
SELECT comm.name, contname, cont_date, amount
FROM comm INNER JOIN contMN ON comm.ID = contMN.RECIP_ID
where contname like "ciresi, m*"
```

We can see that he did donate to other campaigns, but it looks like quite a bit went to his own campaign. Now we can summarize to get numbers we could put in a story:

```
SELECT comm.name, sum(amount)
FROM comm INNER JOIN contMN ON comm.ID = contMN.RECIP_ID
where contname like "ciresi, m*"
group by comm.name
order by 2 desc
```

We can start making the queries a bit more complex. Let's limit our answer to only records where the contribution went to "AL FRANKEN FOR SENATE"

```
SELECT CONTNAME, CITY, CONT_DATE, AMOUNT, NAME
FROM comm INNER JOIN CONTMN ON comm.ID = CONTMN.RECIP_ID
WHERE name="AL FRANKEN FOR SENATE"
order by amount desc
```

Does this tell us who the largest donor was to this committee?

Answer: No. It tells us who wrote the biggest single check. Each record represents one donation and it's possible that someone could donate more than once. In order to find the largest donor, we need to do a group by query.

Setting up a group by query with a join can sometimes be confusing, but you just have to remember that the query is exactly the same as if you were doing one table, EXCEPT for the join statement in the FROM line.

So first step is to think about what you want your answer to look like. In this case we want a list of contributor names – one record for each contributor – and a total of the money they donated. Note: Access is going to want you to include the table name with the ZIP field because both tables have a field called "zip"; so the computer needs you to specify which table you want the field to be pulled from. In this case, we want the zip code of the donor, so we need to pull it from ContMN table. You do that by putting the name of the table with a period in front of the field name. Example: "contmn.zip"

```
SELECT contname, contmn.zip, sum(amount)
FROM comm INNER JOIN Contmn ON comm.ID = CONTMN.RECIP_ID
WHERE name="al franken for senate"
group by contname, contmn.zip
Order by 3 desc
```

Let's change the order of this query so we can look for inconsistencies in the CONTNAME field.

```
SELECT contname, contmn.zip, sum(amount)
FROM comm INNER JOIN CONTMN ON comm.ID = CONTMN.RECIP_ID
WHERE name="al franken for senate"
group by contname, contmn.zip
Order by 1
```

Right at the top we see that we've got dirty data. It shows "Aafedt, Michael" in one record and "Aafedt, Michael D" in another – both with the same zip code. UGH. So that means the query we ran above showing the "largest donor" is probably not accurate. We would need to do some serious data cleanup to do that for a story.

We'll have to leave that for another lesson.

Instead, let's move on and look a little closer at the Al Franken-Norm Coleman Senate race of 2008. As you probably recall, Coleman was initially called the victor but the race was too close and a recount was required by law. The recount turned into a long, drawn-out court battle that took the better part of 2009 to resolve. In the end, nationally-known comedian Al Franken narrowly edged out the incumbent Coleman.

This database includes all of the contributions these two campaigns – including their Recount committees – got from people in Minnesota and around the country. Let's see who won the money battle.

If you do a query on the committee table looking for these two, you'll discover that they both had multiple committees.

One way we can see a list is by joining the COMM table to the CAND table. Note: the COMM table has a field toward the end called "cand_ID" that matches to the ID field in CAND

```
SELECT comm.NAME, cand.NAME
FROM comm INNER JOIN cand ON comm.CAND_ID = cand.ID
WHERE (cand.name like "coleman*" or cand.name like "franken*") and mailstate="MN"
```

So if we want to look at all the money flowing into their coffers, the easiest way for us to get that is to join all three tables together and just have it look for either "Coleman, Norm" or "Franken, Al"

Remember how a join works: it will only pull records where it finds a match. So if we join all three tables together – and not limit the query in any way with a WHERE statement – then it will

return contributions that only went to committees that are tied to a candidate. It won't tally up or include donations made to other committees (such as the Republican National Party or some political action committees).

So we could tell it to show us all contributions that Norm Coleman received by doing this:
 SELECT comm.NAME, contMN.*
 FROM (contMN INNER JOIN comm ON contMN.RECIP_ID = comm.ID) INNER JOIN cand ON
 comm.CAND_ID = cand.ID
 WHERE (((cand.NAME)="coleman, norm"))

That query gives us 13,666 records (which does include some refunds, which we'll talk about later) but it shows which committee the money went to, along with all of the fields from the ContMN table such as contributor name, amount, date, etc.

Now that we have that join in place, we can leave that and just change the select, where, group by and order by lines of our SQL to ask different questions.

How much did each candidate raise in total? This query will tally up all the money, regardless of which committee it went to:

```
Select cand.name, sum(amount)
FROM (contMN INNER JOIN comm ON contMN.RECIP_ID = comm.ID) INNER JOIN cand ON
comm.CAND_ID = cand.ID
where cand.name= "coleman, norm" or cand.name= "franken, al"
group by cand.name
order by 2 desc
```

It looks like Coleman edged out his opponent in the money race!
 Copy the results of this query into Excel. We'll come back to that in a moment.

How much of that came from Minnesota voters? We just need to tweak the WHERE line of our SQL so that it only counts donations that came from people/organizations with Minnesota addresses. Note the use of parentheses around the OR phrase to pull the two candidates, so that it's separate from the AND phrase to grab the contributor state.

```
SELECT cand.name, sum(amount)
FROM (contMN INNER JOIN comm ON contMN.RECIP_ID = comm.ID) INNER JOIN cand ON
comm.CAND_ID = cand.ID
WHERE (cand.NAME="coleman, norm" or cand.name="franken, al") and contmn.state="MN"
group by cand.name
order by 2 desc
```

Also put the results of this query in the same Excel worksheet where you put the previous query results — then calculate what percentage of each candidate's funds came from Minnesota voters.
 Yikes! Big difference...

name	TOTAL	MN Voters	Pct
COLEMAN, NORM	\$ 11,609,675	\$ 4,623,646	40%
FRANKEN, AL	\$ 10,999,451	\$ 1,873,534	17%

Now let's look at who gave Coleman the most money?
 SELECT contname, contmn.zip, Sum(CONTMN.amount)
 FROM (cand INNER JOIN comm ON cand.ID = comm.CAND_ID) INNER JOIN CONTMN ON
 comm.ID = CONTMN.RECIP_ID
 WHERE cand.NAME = "coleman, norm"
 group by contname, contmn.zip
 order by 3 desc

Sally W. McGuire comes to the top with \$4,000. Considering the problems we found earlier with contributor names, we might want to double check this is the correct amount for Sally.

Let's start with a BROAD query that will look for anyone named McGuire who gave to Norm Coleman:

```
SELECT contname, contmn.zip, cont_date, amount
FROM (cand INNER JOIN comm ON cand.ID = comm.CAND_ID) INNER JOIN CONTMN ON
comm.ID = CONTMN.RECIP_ID
WHERE cand.NAME = "coleman, norm" and contmn.last like "mcguire*"
```

That gives us what appears to be several different people. But we can see that the zip code field is pretty consistent for the records for Sally. So let's limit it a bit:

```
SELECT contname, contmn.zip, cont_date, amount
FROM (cand INNER JOIN comm ON cand.ID = comm.CAND_ID) INNER JOIN CONTMN ON
comm.ID = CONTMN.RECIP_ID
WHERE cand.NAME = "coleman, norm" and contmn.last like "mcguire*" and
contmn.zip="55439"
```

Now we can see that it looks like James McGuire might be her husband (we'd have to do some outside research to confirm), and we can also see a couple other things. First, some dirty data. Note: SALTY instead of Sally. And we've got a couple refunds. Both James and Sally paid in \$2,300 that were refunded (note the cont_date on the original payment and the refund will be the same).

When we're summing up the money, the refunded donations will be offset by those refunds, so we should get the correct amount (note: this dataset is the final version at the end of the cycle. If you get campaign finance data midway through the cycle sometimes the refund doesn't show up until the next quarter).

But the "Salty McGuire" record was not counted in our original total. So instead of \$4,000, it looks like Mrs. McGuire really gave \$6,600.

Can we truly say she was the largest donor? Nope. Not until we clean up the contributor names.

```
-----
Who was Franken's biggest contributor?
SELECT contname, contmn.zip, Sum(CONTMN.amount)
FROM (cand INNER JOIN comm ON cand.ID = comm.CAND_ID) INNER JOIN CONTMN ON
comm.ID = CONTMN.RECIP_ID
WHERE cand.NAME = "franken, al"
group by contname, contmn.zip
order by 3 desc
```

The name that comes to the top of this list sounds awfully familiar. Can we figure out who it is based on other information in the table? Perhaps.

Let's try this:

```
SELECT comm.name, contMN.contname, contmn.city, occupation, cont_date, amount
FROM contMN INNER JOIN (comm INNER JOIN cand ON comm.CAND_ID = cand.ID) ON
contMN.RECIP_ID = comm.ID
WHERE cand.NAME="franken, al" AND contMN.CONTNAME Like "myers, m*"
```

Occupation field says.... "self-employed/Actor"
Hmm. THE Mike Myers....as in the guy from Saturday Night Live? And Austin Powers? And Wayne's World?

Note that we can see he came through with two big checks right before the election. However the checks are the same amount and to the same committee. I'd be a little suspicious. And the two \$2,300 checks to his campaign committee — on the same date — raise my eyebrows a bit. Is it possible the campaign double entered one donation?

Could there possibly be other famous actors who gave to Franken?

Let's find out.

```
SELECT comm.name, contMN.contname, contmn.city, occupation, cont_date, amount  
FROM contMN INNER JOIN (comm INNER JOIN cand ON comm.CAND_ID = cand.ID) ON  
contMN.RECIP_ID = comm.ID  
WHERE cand.NAME="franken, al" AND contMN.occupation like "**actor**"
```

Time for some shoe-leather reporting to check out what we found...

Created by MaryJo Webster
St. Paul Pioneer Press
maryjo.sylwester@gmail.com
Updated June 2011