Preliminary: Control-Click this link, [**recycle 2010.accdb**](http://www.777rauer.com/accesstest/recycle%202010.accdb)**,** to open up the access database. Enable contents and save to your system. This should put the recycle 2010.accdb database on your system. You will be creating 5 queries within this database and this database will be sent back to me by the time indicated. If it arrives later than the deadline it will not count.

Only your work will be accepted

1. Open up the **recyclecommercia**l table of the [**recycle 2010.accdb**](http://www.777rauer.com/accesstest/recycle%202010.accdb) database. You can find this database on my web site but it is easier to load it through the link.. Notice that among the field column names is **C13** and **C14**. Now, look also at the table, **recycleid**. Write a query (and I will only accept this and the rest of what's indicated as a query that I, myself, can run off your database) that prints out Designation, Recycle name, percent and definition for the designations **c13** and **c14**. Note, this query should involve a logical ***or*** to do this. Save this query as **q1**.
2. For your test, we are going to concentrate on the material **C13** which is some material that you can see by looking at recycleid. Write a query that displays the tonnage of each row (record) in **recyclecommercial** for the material c13 if the value is more than zero. Remember that conditions such as this, greater than zero, is a **criteria in QBE** and a **where** in SQL. Show **idno, townshipid, generatorid** and the amount in **C13**. The query should display these records in descending order of **C13 (this is sorting)**. Save this query as **q2**.

In the **generator** table is the name of the generators of recycled material. Notice there is a name associated with a number designated as **generatorid**. Notice that there is a **generatorid** designation in **recyclecommercial.** This will be our ultimate inner join. But we need to get to it by a set of queries.

1. Create a query that gives the total (sum) of **C13** for those records in **recyclecommercial** where c13 is greater than or equal to 13. This is an aggregate function using the sum function. This will give you one number. Save this query as **q3**.
2. Create another query that does something similar to the **q3** query (including the need to use only those records that contain 13 or more tons of C13) except that subtotals by **townshipid** are displayed. For this, you will need to modify q3 (remember to rename this), adding **townshipid** next to your sum for display. But, SQL will not allow this without one additional command. At the end of your SQL statement (or using **group by** in QBE) you will need to add the text ***group by townshipid*** . (remember, except for group by, detail and aggregate do not mix) Save this as **q4**.
3. Now, we want to show all the generators of at least 13 tons of **C13**. We will use the link between **generatorid** in **recyclecommercial** and **generatorid** in **generator**. This is an inner join. So, linking these two tables (it's easiest to do this in QBE), create a query indicating the name of the generator and the amount of **C13** recycled. List these names and the number of tons by descending amount of **c13**. Consider what you did above in the query **q4**. This is another example of the need to use **group by** although this time on generator name (***group by generator.name***). Save this query as **q5**

Send this database (revised by your 5 queries) as an attachment to [777rauer@voicenet.com](mailto:777rauer@voicenet.com). If possible, add you name to the access database name.