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 **tonnage**. In addition you will see townshipid, subhaulerid and generatorid. Now, look also at the table, **Townships**. Write a query (and I will only accept this and the rest of this test as queries that I, myself, can run off your database when submitted) that prints out Name, population, county for the townshipids of **12 and 13**. Note, this query will involve a logical ***or*** to do this. Save this query as **q1**.
2. For your test, we are going to concentrate on the township 12 (t**ownshipid**=12) which is East Norriton Township. Write a query that displays the tonnage of each row (record) in **recyclecommercial** for the field (column) **tonnage** for townshipid=12. Remember that conditions such as **townshipid=12** is a **criteria in QBE** and a **where** in SQL and you are using this throughout these questions. In addition, for each row(record) show **idno, generatorid** and **tonnage**. The query should display these records in descending order of **tonnange**. 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 tonnage for those records in **recyclecommercial** where **townshipid = 12**. In addition, we only want to show those records with **tonnage** greater than or equal 26. 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 greater than or equal 26 in tonnage and townshipid=12) except that subtotals by **subhaulerid** is displayed. For this, you will need to modify q3 (remember to rename this), adding **subhaulerid** 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 subhaulerid*** . (remember, except for group by, detail and aggregate do not mix) Save this as **q4** but also save this as an excel spreadsheet. Call the excel spreadsheet **q4**
3. Now, we want to show all the generators of equal or at least 26 tons of **tonnage** 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 **tonnage** recycled. List these names and numbers by descending amount of **tonnage**. 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** but also send this out as an excel spreadsheet. Call the excel spreadsheet **q5**.

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