

<http://www.cs.arizona.edu/classes/cs460/fall08/>

Homework #3: SQL Queries

Due Date: October 30st, 2008, at the beginning of class

System Requirement:
You MUST use our local Oracle DBMS installation for this assignment.

Overview: This assignment is simply meant to give you the opportunity to get some practice with the formulation of SQL queries. The DBMS we'll be using is the Oracle Database 10g Enterprise Edition, the not-quite-latest version of the database system first created in 1978. We won't be using many Oracle-specific features in this assignment; the goal is to give you some practice formulating and testing basic SQL queries.

Software: Oracle 10g runs on a machine in our department named "aloe," but we will access it from lectura. You should each have an account on Oracle. The username is your lectura username, and your password is the letter 'a' concatenated to the last 4 digits of your CSID (e.g., a3456 if your CSID is 123456). There are two ways to access Oracle's command-line querying program, SQL*Plus:

1. On lectura, a script named `sqlpl` exists that makes the connection from lectura to aloe for you. To run it, type the following in a lectura terminal window, replacing 'username' with your lectura login name:
`sqlpl username@oracle.aloe`
You will then be prompted for your "a####" password.
2. From a web browser, you can use this URL:
`http://aloe:5560/isqlplus`
You will be prompted for your username, password, and a "connect identifier", which is `oracle.aloe` as in the first option.

I've set the tables of the aquarium database (from Homework #2) and the Supplier-Part-Project database to be accessible by you. In addition, you can create your own tables to play with. I strongly suggest that you attempt to access Oracle ASAP to verify that your Oracle access was set up correctly. First, connect to Oracle as shown above. At the SQL*Plus prompt (`SQL>`) type this query: `select * from mccann.species;` (don't forget the "mccann." and the semicolon!). If you see the content of the species table, all should be well.

Assignment: Basically, the assignment is to redo most of the queries you answered in Homework #2 (with a few substitutions/additions) using SQL. I've created tables that contains the same information as the LEAP aquarium database. Here is the schema again, with slight changes (in particular, the foreign key names now match the primary key names).

Species (sno, sname, sfood)
Tank (tno, tname, tcolor, tvolume)
Fish (fno, fname, fcolor, fweight, tno, sno)
Event (eno, fno, edate, enote)

Using Oracle and the aquarium database, write SQL queries that answer the following questions. Most are holdovers from Homework #2, but some are new and some are slightly modified. If you find any questions that you can't answer, explain why. (But be aware that I believe all of them to be possible.)

1. What are the names of all of the red fish?
2. What are the colors of all tanks named “lagoon”?
3. What is the Cartesian Product of the sname field from Species with the tname field from Tank? List each (sname,tname) pair only once.
4. What are the colors of the sharks (in alphabetical order)?
5. What is the name of the heaviest fish?
6. What are the names of the fish that are sharks and live in cesspools?
7. The database contains names of species, tanks, and fish. Display a result containing all of these names.
8. What are the names of species found in puddles?
9. What are the names of species that are found in the same tank with a shark? List each species name only once in the result.
10. What are the names of the fish that have been born and are swimming?
11. What are the names of the fish that have been born but are NOT swimming?
12. What are the colors of the fish and the average weight of the fish of each color? Include in your result only those colors (with the associated average weights, of course) that have an average group weight under 40, and list the results in descending order by weight.
13. What are the names of the species that eat herring that have a representative in all green tanks?

My guess is that you will find #9 to be the most challenging. Note that you do not need to create any temporary relations to write any of these queries in SQL (not even #9).

Hand In: Turn in a printout that shows your SQL queries and the answers Oracle produces when it runs them; `script` can capture Oracle’s output. Please produce the answers in the same order as the questions are listed, and clearly number each of your answers (you can do that by hand). Be sure to write your name on the top of the first page of your printout. In addition, ‘turnin’ your queries; the submission name is `cs460h3`.

Want to Learn More About Oracle?

- Oracle documentation (and there’s a **lot** of it!) is available on-line:
<http://www.oracle.com/pls/db102/homepage>
 For this assignment, the SQL documentation is the most useful (see the bottom entry in the “Most Popular” section).
- Oracle has a “free to download, free to learn” policy. So, if you want to download and install Oracle for your personal educational use, you may. See:
<http://www.oracle.com/technology/software/index.html>

Other Requirements and Hints:

- You can easily capture Oracle’s output to a file by running `sqlpl` within the `script` command. Another option is to use SQL*Plus’s `spool` command.
- For set difference, remember that Oracle uses the MINUS operator instead of EXCEPT.
- In Oracle, executing a file of SQL commands from within SQL*Plus uses the same basic syntax as LEAP: `@ filename`. Example: `@ query01.sql` SQL*Plus looks for `filename` in your current directory.
- If you *really* want to create a temporary table to hold the result of a query, you *can* do that in Oracle ... but you probably shouldn’t, for performance reasons. You could create a table in advance to hold results, and then use the `insert into <relation> <select stmt>;` variation of `insert`. However, in this assignment, doing this is not necessary; you can construct all of these queries without manually creating and populating any additional tables. (If you find yourself wanting to do that, you’re probably thinking procedurally, and SQL isn’t relational algebra!) Note that other DBMSes do support temporary tables.
- And finally: Please remember that a correct answer is a query that produces the correct result *in a logically correct way!* Write queries that will work even if the relations’ content changes.