CSc 460 — Database Design
Fall 2020 (McCann)

http://u.arizona.edu/~mccann/classes/460

Homework #3

(100 points)

Due Date: October 22nd, 2020, 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 11g 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 11g runs on a machine in our department named "aloe," but we will access it from lectura. You now have an Oracle account. The username is your lectura username, and your password is the letter 'a' concatenated to the last 4 digits of your CSc 460 grade identifier (e.g., `a3456` if your identifier is 123456).

To access Oracle's command–line querying program, SQL*Plus, start by SSHing to lectura. Then, run a script named `sqlpl` with a command–line argument of this form:

```
sqlpl username@oracle.aloe
```

where 'username' is your NetID.[1] You'll be prompted for your `a####` password, and then you'll get a command prompt, after which you can start typing queries.

I've pre–loaded the relations of the aquarium database (from Homework #2), and of the Supplier–Part–Project database, for you to access. 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 contain the same information as the LEAP aquarium database. Here is the schema again, with slight changes to some of the field names.

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

Field types are easy to determine from the field content. `edate` is just a 5-character string, not an Oracle `date` type.

Using Oracle and the aquarium database, write SQL queries that answer the following questions. Many 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.)

(Continued . . .)

[1] If your NetID includes non–alphabetic characters, run `sqlpl` with no command–line arguments. Then, at the "Enter username:" prompt, logic with double–quotes around your username. E.g., `"netid4me"@oracle.aloe`

1. What are the names of all of the red fish?
2. What are the colors of all of the 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?

Note that you should not create any temporary relations to write any of these queries in SQL; this isn't RA!

Hand In: Create and submit, using `turnin` on lectura, the following items. The submission folder is `cs460h3`.

1. A text file named `netid.txt` (replace 'netid' with your NetID) containing (a) your name, (b) your SQL queries (from your `.src` files), and (c) the answers Oracle produces when it runs them. Please show your queries in ascending numerical order (1 to 13), and produce the answers in the same order.

   Creating this text file is easy if you use the script utility program; see below. Start script, use cat to display your .src files, use Oracle to produce your query results, exit script, and rename the resulting file.

2. Your Oracle query .src files (as a `tar` file).

Want to Learn More About Oracle?

- Oracle documentation (and there's a **lot** of it!) is available on–line:
  `http://docs.oracle.com/cd/E11882_01/index.htm`
  Be aware that it isn't likely to be very useful for this assignment.
- Oracle has a free 18c Express Edition, if you want to play with it (I've never looked at it). See:
  `https://www.oracle.com/database/technologies/appdev/xe.html`

Other Requirements and Hints:

- As noted above, 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` (e.g., `@ query01.sql`) SQL*Plus looks for `filename` in your current directory.
- You really, *really* do NOT want to create temporary tables for this assignment, because that's a sign that you're thinking procedurally (like RA) instead of non–procedurally (like TRC). That said, it is possible to save query results into tables in Oracle, but doing so is discouraged for performance reasons. You would create a table in advance to hold results, and then use the `insert into <relation> <select stmt>;` variation of `insert`. Other DBMSes support truly temporary tables that provide better performance ... and you probably wouldn't want to use them, either.
- 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.
- Finally, and as always: Start early!