

Homework #3: SQL Queries

Due Date: October 18th, 2018, at the beginning of class

Name: _____

Directions: See the next two pages for the homework instructions.

On or before the due date, by the start of class, hand-in a printout of your solutions with this cover page stapled to the front **and** submit your electronically-formatted version of your solutions (the turnin folder is **cs460h3**). If you need to submit your solutions within the 24-hour late window, place your printout in the CS 460 mailbox (bottom row) in CS 713 as soon as you are able to do so.

— Points —

Question # Possible Deductions

1	5	
2	5	
3	5	
4	5	
5	10	
6	5	
7	5	
8	5	
9	10	
10	10	
11	10	
12	10	
13	15	
Totals:	100	

Score:

(Continued ...)

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 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 class grade identifier (e.g., a3456 if your ID 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. You'll be prompted for your password, and then you'll get a prompt.

I've set the tables of the company 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.dependent;` (don't forget the "mccann." and the semicolon!). If you see the content of the dependent 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 company database. Here is the schema again, with slight changes to some of the field names:

```
Employee (fname, minit, lname, ssn, bdate, address, sex, salary, superssn, dno)
Dept.Locations (dnumber, dlocation)
Department (dname, dnumber, mgrssn, mgrstartdate)
Works.On (essn, pno, hours)
Project (pname, pnumber, plocation, dnum)
Dependent (essn, dependent_name, sex, bdate, relationship)
```

Field types are easy to determine from the field content. Dates are stored using Oracle's date type.

Using Oracle and the company database, write SQL queries that answer the following questions. 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. List all attributes of the employees from department 5.
2. What are the SSNs of the employees working on proposal #10? Sort the result in descending order.
3. What is the Cartesian Product of the employees' first names and the dependents' names? List each pair of names only once in the result.
4. At which locations is the research department located?
5. What is the full name of the employee who has the largest salary?
6. What are the names of the departments with employees who have a dependent named Alice?
7. Retrieve the SSNs of the employees who work in department 5 or who directly supervise an employee in department 5.

(Continued ...)

8. What are the birthdates of the managers of the departments with projects from Stafford? (Note the wording change from Homework #2.)
9. For each employee, retrieve his/her full name and the full name of his/her immediate supervisor.
10. What are the names of the male dependents who are dependents of a male employee?
11. What are the salaries of the employees from department 5 who are NOT working on ProductY?
12. For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.
13. What are the names of the departments which employ all genders?

Note that, unlike LEAP, you do not need to create any temporary relations to write any of these queries in SQL.

Hand In: (1) 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). Remember to staple this handout to the front of your answers, and be sure to write your name on the top. (2) Submit your SQL queries (as one big `.sql` file, or as a tar file of separate `.sql` files) using `turnin`. The submission folder is `cs460h3`.

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 11g 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:

- 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.