Directions: 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 GS 713 as soon as you are able to do so.

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-exactly-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 should each have an account on Oracle. The username is your lectura username, and your password is currently a1234. Please change it with the ‘password’ command on your first login.

To access Oracle’s command-line querying program, SQL*Plus, start by SSHing into lectura. Then, run the sqlpl script with a command-line argument of this form:

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

where your ‘username’ is your NetID. You’ll see a prompt for your password, and after entering it, you’ll see Oracle’s SQL*Plus prompt (SQL>). (Note: If your NetID contains non-letter characters, run sqlpl without an argument. When prompted for a login, type in "username"@oracle.aloe)

I’ve pre-loaded the relations of the safari database (from Homework #2), as well as those of the Supplier–Part–Project database, for 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, type this query: `select * from mccann.sale;` (don’t forget the “mccann.” and the semicolon!). If you see the content of the sale 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 and additions) using SQL. I’ve created tables that contain the same information as the LEAP safari database. Here is the schema again, with slight changes to some of the field names.

```
Sale (saleno, saleqty, itemno, dname)
Supplier (splno, splname)
Item (itemno, itemname, itemtype, itemcolor)
Department (deptname, deptfloor, deptphone, empno)
Delivery (delno, delqty, itemnum, dptname, splno)
Employee (empno, empfname, empsalary, departname, bossno)
```

Using Oracle and the safari database, write SQL queries that answer the following questions. (Remember, there are changes from Homework #2, so don’t just redo that list, use the following one instead.) If you find any questions that you can’t answer, explain why. (But be aware that we believe all of them can be answered.)

1. What are the names of the suppliers?
2. What are the names of the employees in Marketing?
3. What is the Cartesian Product of the suppliers’ names and the departments’ names?
4. What are the item numbers of the items sold by the departments located on the second floor? (Use \(\bowtie\).)
5. What are the item numbers of the items sold by the departments located on the second floor? (Use nested selects instead of \(\bowtie\).)
6. Find the ID numbers and names of those employees who earn more than their manager.
7. List the numbers of the items delivered by Nepalese Corp or sold in the Navigation department.
8. What are the names of the items sold on floors other than the second floor?
9. Find the names of the items sold by no department on the second floor.
10. List the names and salaries of the managers who manage more than two employees.
11. What are the names of the suppliers of Pith helmets sold in a department managed by Andrew?
12. For each department on the second floor, find the average salary of each department’s employees.
13. What are the item numbers of the items sold by all departments on the second floor? (Use \(\div\))

Note that, unlike in LEAP, you do not need to create any temporary relations to write any of these queries in SQL. See also the Other Requirements and Hints section, below, for a few additional thoughts.

(Continued . . . )

---

1If your NetID includes numbers, run sqlpl with no arguments. At the “Enter user-name:” prompt, login with double quotes around your username. E.g., "netid123"@oracle.aloe

2“Don’t Forget The McCann” is good advice in most situations. ©
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.

- Submit your SQL queries (as one big .sql file, or as a tar file of separate .sql files), using ‘turnin.’ The submission folder name 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/
  For this assignment, the “SQL Language Reference” is likely to be the most useful (it’s in the “Supporting Documentation” section).

- Oracle offers a free 11g Express Edition. I’ve never downloaded it, but if you want to play with it, visit: http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/

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.

- 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.

- Seems like I’m forgetting something ...Oh, right: Start early!