

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

Homework #3: SQL Queries

(100 points)

Due Date: March 16th, 2023, at the beginning of class

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 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 the SQL*Plus prompt (`SQL>`). (You can change your password, with the 'password' command, but there's no real reason to, as you won't be storing any sensitive data.)

I've set the tables of the ebook 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>` prompt, type this query: `select * from mccann.book;` (don't forget the "mccann." and the semicolon!). If you see the content of the book 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 ebook database. Here is the schema again, with slight changes to some of the field names:

```
book (isbn13,title,edition,category,price,copyright,pages,pid)
encoding (isbn13,format,drm)
writer (wid,surname,givenname,city,state,zipcode,phone,email)
publisher (pid,name,address,city,state,zipcode,url)
authorship (isbn13,aid,percentage)
review (rid,isbn13,stars)
```

Field types are easy to determine from the field content. Copyright dates are just stored as integers.

Using Oracle and the ebook 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.)

(Continued ...)

1. What are the names of the publishers?
2. What are the full names (given and surnames) of the writers who live in California (CA)?
3. What is the Cartesian Product of the writers' cities and the publishers' cities? Show only the unique pairings.
4. (You **must** use JOIN for this query.) What are the titles of the ebooks encoded in azw3?
5. (You must use nested selects for this query, instead of join.) What are the titles of the ebooks encoded in azw3?
6. What are the surnames of the writers who have at least one of their books available as a PDF? Display each name only once, and in alphabetical order.
7. What is the title and price per page of the book with the largest price per page? Your result must consist of a single tuple.
8. What are the full names of the writers who have written one or more ebooks as a coauthor?
9. Display the titles of the books that have no encodings.
10. What are the full names, and quantities of books written, of the writers who have authored (including coauthoring) at least two books?
11. What are the names of the publishers of the books written by book authors who are also reviewers?
12. List, in ascending order, the IDs of all of the writers, and, for each, the ISBNs of the books that they have reviewed.
13. What are the IDs of the reviewers who have reviewed all of Chousky's books? (Yes, this is the \div query.)

Note that, unlike LEAP, you do not need to (and should not; see below) create any temporary relations to write any of these queries in SQL.

Hand In: (1) Create a `typescript` file as follows: Start `script` (see below), use the `cat` command to display your query file(s) in ascending order (1-13), run your queries in Oracle (in the same order), and exit `script`. (2) Use `turnin` to submit your `typescript` file and your query `.src` file(s) to the `cs460h3` folder.

Want to Learn More About Oracle?

- Oracle documentation (and there's a **lot** of it!) is available on-line:
https://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. When you are done capturing the screen output, type `exit`. The collection of captured text will be found in a file in the current directory named `typescript`. 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`. SQL*Plus looks for `filename` in your current directory. Example: `@ query01.sql`
- 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 both philosophical and 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.