

Applying SQL #4

(estimated time: 120 minutes, 35 pts)

Background

This exercise has you use material you've seen in chapters 7 and 8 of our Oracle SQL text. You will use objects in both the Issue25 and Student schemas. You will find it helpful to use the schema handouts as a reference.

Tasks

1. Launch *SQL*Plus* and login to our *cis119do* database.
2. Spool the results of your session to a file.
3. Issue the following commands:

```
SQL> set pagesize 50
SQL> set linesize 120
SQL> select user, to_char(sysdate,
'MON-DD hh:mi am') from dual;
```
4. Recall that every table in the book's Student database schema has a field named `MODIFIED_BY` which indicates the user who last modified the record. To become familiar with the values used in the next few steps, show an *unduplicated* list of users who have modified a record in the `GRADE` table. Show only the `MODIFIED_BY` field.
5. Now, show an *unduplicated* list of users who have modified a record in the `SECTION` table. Show only the `MODIFIED_BY` field.
6. Using *set operators*, show users who have modified a record in *both* the `GRADE` table and the `SECTION` table. Suppress duplicate rows. Show only the `MODIFIED_BY` field.
7. Using *set operators*, show users who have modified a record in *either* the `GRADE` table or the `SECTION` table (or both). Show only the `MODIFIED_BY` field and show each value only once (suppress duplicates).
8. Using *set operators*, show users who have modified a record in the `GRADE` table but *not* in the `SECTION` table. Show only the `MODIFIED_BY` field.
9. Use a subquery to determine the `WRITERID` of writers who have written more than 2 articles, then have an outer query show the writer's *concatenated* name and phone number. *Do not use an equijoin. Do not use a correlated subquery.*
10. Use a subquery to show the *concatenated* name and amount for any *freelance* writer whose amount is less than the average amount paid to *freelancers*.
11. Show the *concatenated* name and last contact date for writers who have no articles shorter than 2000 words. Accomplish this using a subquery. *Do not use an equijoin.*
12. Show the title, type and length of articles that are longer than *every* `EXP` article. Accomplish this using a subquery. *Do not use an equijoin.*
13. For each course offered, show the course number, description, and the number of sections offered. Sort the results by course number. Use an inline view to calculate the number of sections. *Do not perform an equijoin within the inline view.*

Course Description	Sections Offered
10 DP Overview	1
20 Intro to Computers	4
25 Intro to Programming	9
100 Hands-On Windows	5
120 Intro to Java Programming	6
122 Intermediate Java Programmin	5
124 Advanced Java Programming	4
125 JDeveloper	5
130 Intro to Unix	4
132 Basics of Unix Admin	2
. . .	
450 DB Programming in Java	1

28 rows selected.

14. A table in my schema contains details regarding a company's purchase receipts for 2007 and 2008. Use the following to become familiar with the table's structure and some of its rows.

```
SQL> desc ttrollenreceipt

SQL> SELECT *
      FROM ttrollenreceipt
      WHERE ROWNUM <30;
```

15. Develop a statement that displays the following receipt summary.

2007 Count	2007 Total	2008 Count	2008 Total	Count Change	Total Change
20000	10224294.9	25000	12693070.4	5000	2468775.48

Hint: Use an incremental development approach and multiple inline views to break the problem into smaller sub-problems. Start by developing a query that processes the 2007 receipts and shows the receipt count and total amount for that year. Then copy that query and modify it to do the same for the 2008 receipts. Then pull together an overall statement that employs multiple inline views to process each year's summary and produces the left-most 4 columns displayed above. Finally, add two columns to calculate the changes.

16. Develop a query that shows each ZIPCODE and the number of STUDENTs living within the ZIPCODE. Then modify the query to show only the four ZIPCODES with the *highest* student residency. *Do not use an equijoin.*

17. Turn off spooling.

18. Open your spooled text file in an editor. If necessary select the entire contents of the file and select a non-proportionally spaced font, such as Courier New. If necessary, use landscape orientation or a smaller font size (eg: 8 point) so lines don't wrap.

19. Next, type the task number (4-14) to identify each task. Also, delete erroneous commands and their resulting output. Do not enter additional commands or simulated output.

Save and print your text file.