

Applying SQL #7

(estimated time: 120 minutes, 45 points)

Background

This exercise employs material from Chapters 12-14 of our Oracle SQL text.

Tasks

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

```
SQL> set pagesize 60
SQL> set linesize 120
SQL> SELECT user, TO_CHAR(SYSDATE, 'mon-dd hh:mi am')
        FROM DUAL;
```

4. Create a view named `business_writers_vu` that shows an *unduplicated* list of writers who have written a BUS article. Each writer who has written a BUS article should be shown only once, even if they've written multiple BUS articles. Show each writer's *concatenated* full name, phone number and last contact date. The view must use a non-correlated subquery, not an equijoin. Test your view and refine it until it works correctly.
5. Use the data dictionary to confirm that your `business_writers_vu` view exists and show the *entire* stored `select` statement the view represents. Show only the `business_writers_vu` view.
6. Create a sequence named `countdown_seq` that generates values from 5 down to 1, then resumes at 5 and counts down to 1 (i.e., generates the values 5, 4, 3, 2, 1, 5, 4, 3, 2, 1, 5, 4,...).
7. Use commands to generate the first seven (7) values of your `countdown_seq` sequence. (i.e., generate 5, 4, 3, 2, 1, 5, 4). Hint: generate the first value, then use `/` to repeat the command 6 additional times.
8. [Note: Tasks 8 and 9 must be performed together in the same session.] Suppose you've been hired as a new instructor. Use an `INSERT` statement to add yourself to the `instructor` table. Your statement must use the `instructor_id_seq` sequence to generate the value for your `instructor_id`. It must also automatically determine the current date/time for the `created_date` and `modified_date` columns.
9. [Note: Tasks 8 and 9 must be performed together in the same session.] Now that you've inserted yourself as an instructor, you've been assigned to teach `SECTION_ID` #122. Use an `UPDATE` statement to assign that section to you. The `UPDATE` statement must use the `instructor_id_seq` to determine the `instructor_id` that was generated when you were inserted into the `instructor` table in the previous task. It must also set `modified_date` to the current date/time.
10. Use the data dictionary to confirm the existence of *each* sequence in your schema. Show each sequence's name and its minimum, maximum, increment, and last-value-generated values.
11. Delete your `countdown_seq` sequence.
12. Create a B-Tree index for *each* foreign key in your `article` table.
13. Use data dictionary views to show details about each index associated with your `article` table. For each index, show its name and uniqueness, along with the field(s) it is associated with. Accomplish this in one statement that employs a join.
14. Use data dictionary views to display, for only the `writer` table, each constraint's name, the field(s) the constraint pertains to, the text of the constraint (if applicable), and, depending on the type of constraint, either the phrase "Primary Key", "Check" or "Foreign Key". Accomplish this in one statement that employs a join.

15. Create an *interactive script* that prompts for and receives a 3-byte code for a type of article, and then displays TITLE, ISSUE and LENGTH for the specified type. The rows should be sorted by TITLE and the substitution details must not be visible to the user, as illustrated below. The script must work whether the user enters the 3-character code in upper, lower or mixed case (eg: BUS vs bus or Bus).

```
Which type of article? Bus
TITLE                               ISSUE    LENGTH
-----
$100 Billion AOL Time Warner Merger Approved  01-JAN-01  1702
AT&T Antitrust Settlement                   01-FEB-82  1600
Building Trade Outlook                      01-APR-84  1437
Dot-Coms Go Bust                            01-JAN-01  1830
The Economy Under Sub-Zero Population Growth  01-DEC-95  1020
Trans-Alaskan Oil Pipeline Opening          01-JUL-77   803
```

16. Run your script using **buS** as the type of article.
17. **Print** the final version of your interactive script.
[Note: if you submit this assignment via email, attach this script file.]
18. Use a command to allow user `ttrollen` to employ `SELECT` statements against your `business_writers_vu` view.
19. Use the data dictionary to identify users who have privileges on your `business_writers_vu` view. Show each user's name and the privilege(s) they possess.
20. Turn off spooling. Open your spooled file and delete erroneous commands and their output.
21. Type the task number (3-19) to identify each command, then save and **print** your file.