

Applying SQL #6

(estimated time: 90-120 minutes, 45 pts)

Background

This exercise has you use material from chapters 10-11 of our Oracle SQL text. You'll continue to use the STUDENT and ISSUE25 tables. You will create a new PAYMENT table and will use DML commands to insert, update and delete rows. You'll also be the *victim* of row locking.

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 100
SQL> set linesize 120
SQL> SELECT user, current_timestamp FROM dual;
```

4. Assume that you have just finished a conference call with Maria Martinez, Leroy Johnson and Wilhelm Seeger. Use a *single* SQL Statement to update their LASTCONTACT dates to the current date and time.
5. Modify the WRITER table to include an email address. Include a mechanism that ensures that a email address field value has an @ symbol and a dot, in that order (eg: ttrollen@yahoo.com but rejects ttrollenyahoo.com or ttrollen@yahoocom). [HINT: use the LIKE operator.]
6. Now test that *Oracle* prevents an invalid email address. Attempt to set Kelly Cox's email to **kcoxhotmail.com** (omitting the @ symbol).
7. Edit the previous statement to set Kelly Cox's email to **kcox@hotmail.com**.

PAYMENT	
CHECKNUM (PK)	NOT NULL
AMOUNT	NOT NULL
WRITERID (FK)	NOT NULL
ISSUE_DATE	NOT NULL

8. Create a new table named PAYMENT, as illustrated. Use an appropriate *data type* for each field. Each *constraint* must have an appropriate name. Declare WRITERID as a foreign key so *referential integrity* will be enforced. When no ISSUE_DATE is provided, store the current system date and time. Do *not* cascade the relationship. If you have trouble, use DROP TABLE OR ALTER TABLE.

9. Add the following payments to your new table:

CHECKNUM	AMOUNT	WRITERID	ISSUE_DATE
1001	350.00	C200	9/16/2008
1111	1.99	C200	9/22/2008
1024	1,200.56	H400	9/20/2008
1037	49.95	S260	9/21/2008

10. Show the rows of your PAYMENT table. Display all four digits of the year.
11. Check number 1024 was actually written for \$120.56 (not 1200.56). Issue a command that will modify the row to correct this error.
12. Issue a command to permanently save the result of the previous DML commands.
13. Enter another payment:


CHECKNUM	AMOUNT	WRITERID	ISSUE_DATE
1002	100.00	J525	9/16/2008



14. Now, attempt to insert check 1002 *again* (just press / if the INSERT command is still in the buffer). You should get an error message. What does the error message mean? **Hand-write your answer now. You'll type it in step 40.**
15. Enter one more payment:

CHECKNUM	AMOUNT	WRITERID	ISSUE_DATE
1055	400.00	B463	9/16/2008

You should receive an error message. Read the error message carefully, and then decide what the error message means. **Hand-write your answer now. You'll type it in step 40.**

16. Check number 1111 was actually issued to writer J525. Update the row but **don't** COMMIT it. Oracle will place a *lock* on the row.

Assignment continues 

17. Launch *second* copy of *SQL*Plus* and login again. You now have two Oracle sessions going. Your *Taskbar* has two  icons, indicating that two instances of *SQL*Plus* are running. The right-most  *Taskbar* icon appears selected (depressed and white) because that's the session you're in now. During the next few steps you'll jump back and forth between these *SQL*Plus* session windows.
18. Remain in the **RIGHT** session and use a **SELECT** statement to show all fields for check number 1111. What **WRITERID** is displayed? Is this **RIGHT** session able to see the uncommitted update that the **LEFT** session is making? **Hand-write your answers now. You'll type them in step 40.**
19. Recall that the **LEFT** session still has an *uncommitted* update and has a lock on the row for check number 1111. Remain in the **RIGHT** session and try to delete the row for check number **1111**. Your **RIGHT** session will appear to *freeze* since it is attempting to delete a locked row. *Don't worry.*
20. Go into the **LEFT** session and **COMMIT** the pending update.
21. Return to the **RIGHT** session. Is this session still *frozen*? What do you see? **Hand-write your answers now. You'll type them in step 40.**
22. Exit the **RIGHT** session and return to the only remaining (previously **LEFT**) session.
23. Delete all of the **BUS** articles. Don't worry... you'll soon undo this change.
24. Show the title and length of each **BUS** article. Of course, this query shouldn't return any rows since you deleted the **BUS** articles in the previous step.
25. Issue a **ROLLBACK**.
26. Once again, show the title and length of each **BUS** article. What happens now? **Hand-write your answer now. You'll type it in step 40.**
27. **Before you continue** ... be sure you saw the message **Rollback complete**. This reverses the effect of the recent **DELETE** from step 23. This is important since the statement you issue in the *next* task will **automatically commit** any pending transaction.
28. Create a new table named **ARTICLE_ARCHIVE** by selecting all columns from the **ARTICLE** table for any articles published prior to the 1990's. Do *not* remove these same rows from the **ARTICLE** table.
29. To confirm the previous command worked, use **DESCR** and **SELECT** commands to show the structure and data contained in your new **ARTICLE_ARCHIVE** table.
30. Delete your **ARTICLE_ARCHIVE** table.
31. Attempt to select from the (now-deleted) **ARTICLE_ARCHIVE** table.
32. List the objects contained in your Recycle Bin.
33. Restore the **ARTICLE_ARCHIVE** table from the Recycle Bin.
34. Once again, attempt to select from the (just-restored) **ARTICLE_ARCHIVE** table.
35. Empty the Recycle Bin.
36. Delete your **ARTICLE_ARCHIVE** table (again), but this time do not allow it to be placed in the Recycle Bin.
37. Turn off spooling.
38. Open your spooled text file and delete any erroneous commands and their resulting output.
39. Now, type the task number (4-36) to identify each command. The commands you issued in the **RIGHT** session won't appear in this text file... it was spooling only the **LEFT** session.
40. Now type your answers for tasks 14, 15, 18, 21, and 26 beneath the output (or error message) from the respective command.
41. Save and **print** your edited text file.