DBS501 Lab4 due on Friday, November 13th by 9pm

Just E-mail submission with Subject like : *501 Lab4 by Smith, John*

You need to provide BOTH --- Code and Output

1) Write a stored Procedure called **mine** that will accept as Input TWO character parameters: first will be in the Visa Expiry Date format (MM/YY) and second will be either P, F or B (any case). Then it will display what DAY is the Last day of the provided input format and also it will count how many stored Procedures, Functions or Package Bodies you have created in your schema. You need to take care in your Exception section if the Expiry Date has an Invalid format and if some other letter was entered.

Here are the outputs.

EXECUTE mine ('11/09','P')

Last day of the month 11/09 is Monday
Number of stored objects of type P is 7
PL/SQL procedure successfully completed.

EXECUTE mine ('12/09','f')

Last day of the month 12/09 is Thursday
Number of stored objects of type F is 2
PL/SQL procedure successfully completed.

EXECUTE mine ('01/10','T')

Last day of the month 01/10 is Sunday
You have entered an Invalid letter for the stored object. Try P, F or B.
PL/SQL procedure successfully completed.

EXECUTE mine ('13/09','P')

You have entered an Invalid FORMAT for the MONTH and YEAR. Try MM/YY.
PL/SQL procedure successfully completed.

2) Write a stored Procedure called **add_zip** that will accept as Input THREE parameters for three columns in the table ZIPCODE (ZIP, CITY and STATE).It will firstly check whether entered ZIP already exists in the database and if YES – it will stop processing with the message. If NOT -- it will insert new row in the table ZIPCODE where other columns will use USER and SYSDATE pseudo columns. Also it will use TWO Output parameters to display message SUCCESS or FAILURE and current # of rows in the table for the entered STATE. Then it will display ALL rows from that STATE. Use BIND variables to display your results. Undo your Insert, when

Success happened. Here are the outputs:

Case 1:

PL/SQL procedure successfully completed.

FLAG
SUCCESS

ZIPNUM
2
INCLUDEPICTURE "http://zenit.senecac.on.ca:5560/isqlplus/cabo/images/t.gif" * MERGEFORMATINET

SELECT * FROM zipcode

WHERE state = 'MI'

ZIP	CITY	STATE	CREATED_BY	CREATED_DATE	MODIFIED_BY	MODIFIED_DATE
48104	Ann Arbor	MI	AMORRISO	03-AUG-99	ARISCHER	24-NOV-99
18104	**Chicago**	**MI**	**DBS501_093A40**	**12-NOV-09**	**DBS501_093A40**	**12-NOV-09**
INCLUDEPICTURE "http://zenit.senecac.on.ca:5560/isqlplus/cabo/images/t.gif" * MERGEFORMATINET						

Rollback completed

Case 2:

This ZIPCODE 48104 is already in the Dataase. Try again.
PL/SQL procedure successfully completed.

FLAG
FAILURE
ZIPNUM
1

SELECT * FROM zipcode

WHERE state = 'MI'

ZIP	CITY	STATE	CREATED_BY	CREATED_DATE	MODIFIED_BY	MODIFIED_DATE
48104	Ann Arbor	MI	AMORRISO	03-AUG-99	ARISCHER	24-NOV-99

3) Re-write the previous question so that you use a stored BOOLEAN FUNCTION called **exist_zip** that will check if the provided zip code already exists in the database or not. Then incorporate your function into the new procedure called **add_zip2.** Outputs remain the same.

4) Write a stored CHARACTER FUNCTION called **instruct_status** that will accept as Input TWO parameters – instructor's First and Last name entered in the Upper case. It will firstly check whether the entered name combination exists, and if NOT – it will stop processing with the message. If YES -- it will then count how many sections is this person scheduled to teach and then display the appropriate message (the basic criteria is more than 9 courses or NO courses or between those two numbers).
You will test your function firstly with the plain SELECT statement (A) and then with the BIND variables (B and C) Here are the outputs:

A) After SELECT statement has been issued

LAST_NAME	Instructor Status
Chow	**This Instructor is NOT scheduled to teach**
Frantzen	**This Instructor will teach 10 course and needs a vacation**
Hanks	**This Instructor will teach 9 courses.**
Lowry	**This Instructor will teach 9 courses.**
Morris	**This Instructor will teach 10 course and needs a vacation**
Pertez	**This Instructor will teach 10 course and needs a vacation**
Schorin	**This Instructor will teach 10 course and needs a vacation**
Smythe	**This Instructor will teach 10 course and needs a vacation**
Willig	**This Instructor is NOT scheduled to teach**
Wojick	**This Instructor will teach 10 course and needs a vacation**

10 rows selected.

B) After INPUT parameters 'PETER' and 'PAN' were provided
PL/SQL procedure successfully completed.

MESSAGE
There is NO such instructor.

C) After INPUT parameters 'IRENE' and 'WILLIG' were provided
PL/SQL procedure successfully completed.

MESSAGE	
This Instructor is NOT scheduled to teach	