Homework 3

1)(20 points) related terms: (procedures, exception,implicit cursor)

write a procedure that accept one parameter (vendor ID) and the procedure print the phone number and the address of the vendor (address1, city, zip code, state). The procedure should check whether the vendor have a phone number registered or not, if not, the procedure should print a message saying "No phone is registered to this vendor". The procedure must handle exception "no\_data\_found". The exception must produce the message "No Data Found, invalid inputs".

Try three test cases and submit their message screenshots:

“create or replace procedure Find\_Phone\_Num(v\_id vendors.vendor\_id%type)

as

v\_phone VENDORS.VENDOR\_PHONE%type;

v\_address1 vendors.vendor\_address1%type;

v\_city vendors.vendor\_city%type;

v\_zip VENDORS.VENDOR\_ZIP\_CODE%type;

v\_state vendors.vendor\_state%type;

begin

select vendor\_phone, vendor\_address1, vendor\_city, vendor\_zip\_code, vendor\_state

into v\_phone, v\_address1, v\_city, v\_zip, v\_state

from vendors where vendor\_id = v\_id;

if v\_phone != NULL then

dbms\_output.put\_line('Phone Number: ' || v\_phone ||', Address: '|| v\_address1 || v\_city || v\_zip || v\_state);

else

dbms\_output.put\_line('No phone is registered to this vendor');

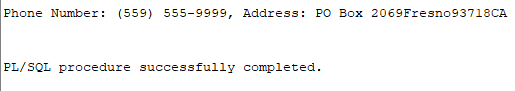
end if;

exception when no\_data\_found then

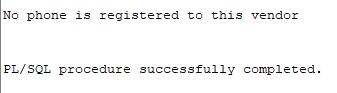
dbms\_output.put\_line('No Data Found, invalid inputs');

end;”

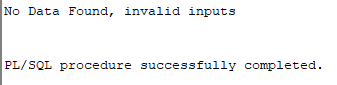
1. Vendor Id: 12



1. Vendor\_id: 120



1. Vendor\_id: 134



2) (30 points) related topics(functions, procedure, count(\*), implicit cursor)

write a function that accept one paramenter (invoice\_id) and return the number of items for that invoice. (FYI: for each invoice we can find the number of items from the table invoice\_line\_items. e.g invoice id=19 has two items). create a procedure that will accept one parameter ( invoice\_id) and the procedure will call the function and print the message " The number of items for invoice\_id : $ is % items" where $ is going to be replaced with the input invoice\_id and % is going to be replaced with the result returned from the function.

Function:

“create or replace function Invoice\_Count(i\_id in invoices.invoice\_id%type)

return number is

return\_count number;

begin

select line\_item\_amt into return\_count from invoice\_line\_items

where invoice\_id = i\_id;

return return\_count;

exception when no\_data\_found then

dbms\_output.put\_line('No Data Found, invalid inputs');

end;”

Procedure:

“create or replace procedure Print\_Invoice\_Count(i\_id invoices.invoice\_id%type)

as

begin

dbms\_output.put\_line('The number of items for invoice\_id : ' || i\_id || ' is ' || Invoice\_Count(i\_id) || ' items.');

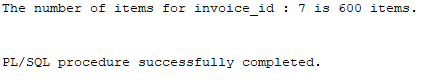
exception when no\_data\_found then

dbms\_output.put\_line('No Data Found, invalid inputs');

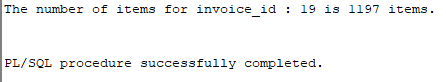
end;”

Try three test cases and submit their message screenshots:

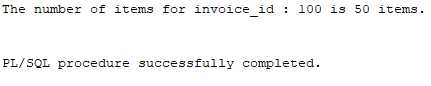
1. Invoice\_id: 7



1. Invoice\_id:19



1. Invoice\_id: 100



3) (20 points) related topics (create table, insert statements, row level trigger, trigger before timing, trigger insert event, new., commit)

create a table called "website\_names" that has three columns:

* Website\_id: number
* Website\_name: varchar(40)
* Website\_create\_date:date

Populate your table with the following data:

1. (1,umbc,10-22-1999)
2. (2,morgan,12-16-2006)
3. (3,towson,06-04-2009)

“insert into website\_names(website\_id, website\_name, website\_create\_date) values (1, 'umbc', TO\_DATE('10-22-1999','mm-dd-yyyy'));

insert into website\_names(website\_id, website\_name, website\_create\_date) values (2, 'morgan', TO\_DATE('12-16-2006','mm-dd-yyyy'));

insert into website\_names(website\_id, website\_name, website\_create\_date) values (3, 'towson', to\_date('06-04-2009','mm-dd-yyyy'));”

Create a row level trigger that fires before the insert operation associated with website\_names table. The trigger displays the new attribute values for website\_name and website\_create\_date

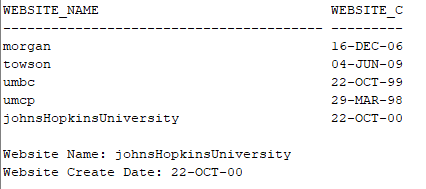
Try two test cases:

1. Insert into website\_names values(5,'johnsHopkinsUniversity', to\_date('10-22-2000','mm-dd-yyyy')

“set serveroutput on;

Insert into website\_names(website\_id, website\_name, website\_create\_date) values (5,'johnsHopkinsUniversity', to\_date('10-22-2000','mm-dd-yyyy'));

select website\_name, website\_create\_date from website\_names;”

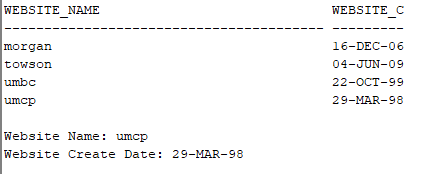


1. Insert into website\_names values(5,'umcp', to\_date('03-29-1998','mm-dd-yyyy')

“set serveroutput on;

Insert into website\_names(website\_id, website\_name, website\_create\_date) values (5,'umcp', to\_date('03-29-1998','mm-dd-yyyy'));

select website\_name, website\_create\_date from website\_names;”



submit the code and screenshot of the messages results.

“SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER insert\_website BEFORE

insert on website\_names

FOR EACH ROW

BEGIN

dbms\_output.put\_line('Website Name: ' || :new.website\_name);

dbms\_output.put\_line('Website Create Date: ' || :new.website\_create\_date);

exception when no\_data\_found then

dbms\_output.put\_line('No Data Found, invalid inputs');

END;”

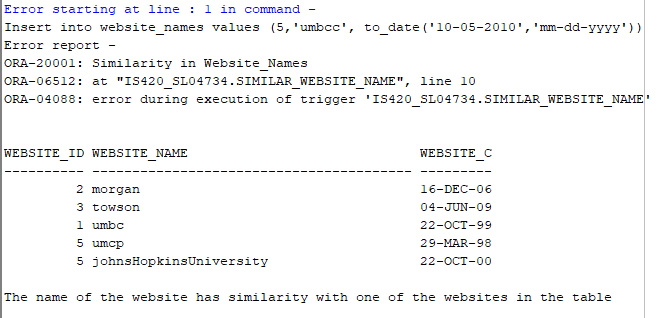
4)(30 points) related topics (insert statements, row level trigger, trigger before timing, trigger insert event, exceptions, explicit cursor,for loop, “LIKE” operation, “%” from week3 slides, commit )

Create a row level trigger that fires before the insert operation associated with website\_names table. the trigger should check if the value of website\_name has any similarity with the names of the websites in the table, if yes, it will print a message that "the name of the website has similarity with one of the websites in the table" and then the trigger will raise RAISE\_APPLICATION\_ERROR(-20001, 'similarity in names');

Try two test cases and take a screenshot of result messages:

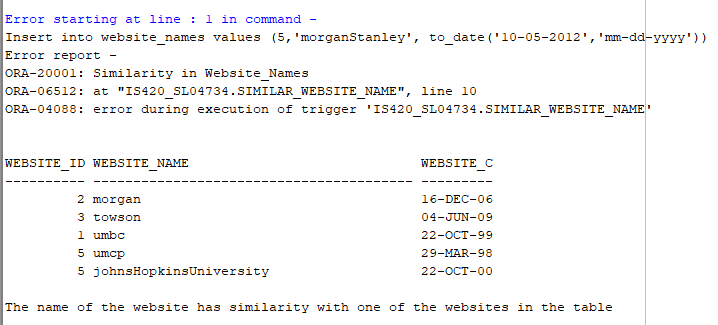
1. “Insert into website\_names values (5,’umbcc’, to\_date('10-05-2010','mm-dd-yyyy'));

Select \* from website\_names;”



1. “Insert into website\_names values 5,’morganStanley’, to\_date('10-05-2012','mm-dd-yyyy'));

Select \* from website\_names;”



Hints:

* you will need a cursor to hold all names in website\_names table
* You will check similarity using 'LIKE' statement and '%websiteName%'
* You will raise application exception to abort insertion upon the similarity using this command line : RAISE\_APPLICATION\_ERROR(-20001, 'similarity in names');
* You will need to add commit to the insert statement to fire the trigger. For example
  + Insert into website\_names values(5,'john hopkins university', to\_date('10-22-2000','mm-dd-yyyy');

commit;

“SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER similar\_website\_name BEFORE

insert on website\_names

FOR EACH ROW

declare

cursor similar\_cursor is

select website\_name from website\_names;

similar\_row similar\_cursor%rowtype;

BEGIN

for similar\_row in similar\_cursor loop

if :new.website\_name like '%' ||similar\_row.website\_name || '%'

then

dbms\_output.put\_line('The name of the website has similarity with one of the websites in the table');

Raise\_application\_error(-20001, 'Similarity in Website\_Names');

end if;

End loop;

END;”