**Problem#1**

Create a stored procedure (exam\_sp) that can be used to insert records into the bb\_shopper table. It should make use of the bb\_shopper\_seq sequence and include IN parameters for firstname, lastname, address, city, state, and zipcode. Include an OUT parameter that will include the Oracle error number if the insert fails.

Include 2 tests, one that shows a successful insert and one that shows a failure. A failure can be caused by passing a value that is too big for a particular field. (for example, state =’ABC’). Test using an anonymous block or host/bind variables. Whichever you prefer.

**Problem #2**

Create a package (exam\_pkg) that will contain the procedure you created in Problem#1. It should be a public procedure. Include a public variable in the package that will be used to record the number of records inserted during the session (ie. The number of times the procedure was executed successfully).

Execute the procedure twice and then prove that the public package variable was updated properly.

1.

create or replace procedure exam\_sp

(p\_firstname in bb\_shopper.firstname%type,

p\_lastname in bb\_shopper.lastname%type,

p\_city in bb\_shopper.city%type,

p\_state in bb\_shopper.state%type,

p\_address in bb\_shopper.address%type,

p\_zipcode in bb\_shopper.zipcode%type,

p\_error out varchar2)

is

begin

insert into bb\_shopper(idshopper, firstname, lastname, city, state, address, zipcode)

values(bb\_shopper\_seq.nextval, p\_firstname, p\_lastname, p\_city, p\_state, p\_address, p\_zipcode);

commit;

exception

when no\_data\_found then

p\_error:= sqlerrm;

when others then

p\_error:= sqlerrm;

end;

/

variable g\_error varchar2(15)

execute exam\_sp('Carla', 'Romero', 'Cleveland', 'OH', '123 Happy Lane', 29005, :g\_error);

select idshopper, firstname, lastname, city, state, address, zipcode

from bb\_shopper

where firstname = 'Carla';

print g\_error

Procedure created.  
  
PL/SQL procedure successfully completed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IDSHOPPER** | **FIRSTNAME** | **LASTNAME** | **CITY** | **STATE** | **ADDRESS** | **ZIPCODE** |
| 30 | Carla | Romero | Cleveland | OH | 123 Happy Lane | 29005 |

execute exam\_sp('Carla', 'Romero', 'Cleveland', 'Ohio', '123 Happy Lane', 29005, :g\_error);

select idshopper, firstname, lastname, city, state, address, zipcode

from bb\_shopper

where firstname = 'Carla';

print g\_error

|  |
| --- |
| **G\_ERROR** |
| ORA-12899: value too large for column "S60"."BB\_SHOPPER"."STATE" (actual: 5, maximum: 2) |

|  |
| --- |
|  |

2.

create or replace package exam\_pkg is

pv\_count number(10);

procedure exam\_pp

(p\_firstname in bb\_shopper.firstname%type,

p\_lastname in bb\_shopper.lastname%type,

p\_city in bb\_shopper.city%type,

p\_state in bb\_shopper.state%type,

p\_address in bb\_shopper.address%type,

p\_zipcode in bb\_shopper.zipcode%type,

p\_error out varchar2);

end;

/

create or replace package body exam\_pkg is

procedure exam\_pp

(p\_firstname in bb\_shopper.firstname%type,

p\_lastname in bb\_shopper.lastname%type,

p\_city in bb\_shopper.city%type,

p\_state in bb\_shopper.state%type,

p\_address in bb\_shopper.address%type,

p\_zipcode in bb\_shopper.zipcode%type,

p\_error out varchar2)

is

begin

insert into bb\_shopper(idshopper, firstname, lastname, city, state, address, zipcode)

values(bb\_shopper\_seq.nextval, p\_firstname, p\_lastname, p\_city, p\_state, p\_address, p\_zipcode);

pv\_count :=pv\_count+1;

commit;

exception

when no\_data\_found then

p\_error:= sqlerrm;

when others then

p\_error:= sqlerrm;

end;

begin

pv\_count := 0;

end;

/

variable g\_error varchar2(15)

execute exam\_pkg.exam\_pp('Carla', 'Romero', 'Cleveland', 'OH', '123 Happy Lane', 29005, :g\_error);

/

begin

dbms\_output.put\_line(exam\_pkg.pv\_count);

end;

Package created.  
  
Package body created.  
  
PL/SQL procedure successfully completed.  
  
Package body created.  
  
1   
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.  
  
Package body created.  
  
2   
PL/SQL procedure successfully completed.