Laboratory 9
Stored Procedures

References
Lecture Notes: Topic 6

In this practice class you will learn how to write procedures and more importantly stored procedures.

Exercise 1 – PL/SQL

a) Type the below code to the Oracle Live SQL or APEX.

```
BEGIN
DBMS_OUTPUT.PUT_LINE  ('Local Events – For all you Event Organising
                      Needs');
END;
/
```

What happens?

b) Type the below code to the Oracle Live SQL or APEX.

```
DECLARE
myString   VARCHAR2(20) := 'Today''s date is: ';
today      DATE;
BEGIN
today := SYSDATE;
DBMS_OUTPUT.PUT_LINE(myString || today);
END;
/
```

What happens?

The above are examples of procedures, however to run them again you have to type in the code again – they have not been named and stored. The rest of this lab looks at writing stored procedures and functions.

Exercise 2 – Stored Procedures

1. Create a stored procedure that takes as its argument the business ID of an event sponsor and displays to the screen the business' name and address. Do you need to use a cursor? Why or why not?

```
SQL> execute sponsorDetails('SP0001');
```

Note: ***in APEX, there are no EXECUTE server command. PL/SQL calls need to be made using anonymous PL/SQL code blocks. Please use below to execute:***

```
begin
  sponsorDetails('SP0001');
end;
```

```
Sponsor Details
---------------
Business Name: Local Town IGA
Address: 25 High St, Local Town 1001

PL/SQL procedure successfully completed.
```

2. Create a stored procedure that displays to the screen the full details of all the promoters. Do you need a cursor? Why or why not?

3. Create a stored procedure that receives a date as input, and displays the event(s) that take place on that day and their start times. When you display the date for the events, also display which day of the week it is.

```
SQL> Execute eventTimesOnDay('09-Oct-2016');
```

Note: ***if in APEX, please use below to execute:***

```
begin
  eventTimesOnDay('09-Oct-2016');
end;
```

```
Events on Tuesday  , OCTOBER   09, 2016:
Event Name: Pearson's Breakfast Meeting
Time: 08:30 am

PL/SQL procedure successfully completed.
```

4. Create a stored procedure that receives an event id as input and displays the number of tickets remaining for each date that the event is on.

```
SQL> Execute ticketsRemaining('E00008');
```

Note: ***if in APEX, please use below to execute:***

```
begin
  ticketsRemaining('E00008');
```

```
end;


Tickets remaining for event id:E00008
Date and time: 24-MAR-2017, 07:30 pm
Number of tickets remaining 180

PL/SQL procedure successfully completed.
```

5. *Local Events* is the name of the company for whom this database has been built. They want to send promotional material to those companies who have not used their services in a while. Create a stored procedure that displays to the screen a mailing list (company name, contact person and address) of all the client companies that have not used the services of *Local Events* for over a year.

NOTE: The result below might be different, depending on when you are running the stored procedures.

```
SQL> Execute lostClients;

Note: if in APEX, please use below to execute:
begin
  lostClients;
end;


Market Hill Council
ATTENTION: Penny Wong
15 Davis St, Market Hill, 1002

Fisherman's Bend Football Club
ATTENTION: Nathan Jackson
70 Mahoney Rd, Fisherman's Bend, 1003

Bayman's Business Group
ATTENTION: Bruce Bayman
Main Rd, Local Town, 1001
```

6. Write a stored procedure that displays the details of the most popular venues and least popular venues so far.

```
SQL> Execute mostAndLeastPopularVenues;

Note: if in APEX, to execute:
begin
  mostAndLeastPopularVenues;
end;


Most popular venue:
Venue Name: Local Town Community Theatre
Street Address: 146 Main Rd
Suburb: Local Town
Postcode: 1001
Least popular venue:
```

```
Venue Name: Town Hall
Street Address: 15 High St
Suburb: Local Town
Postcode: 1001
Venue Name: Local Town Video Conference Centre
Street Address: 140 Main Rd
Suburb: Local Town
Postcode: 1001

PL/SQL procedure successfully completed.
```