Laboratory 10
Stored Functions & Triggers

References
Lecture Notes: Topic 7 & 8

In this practice class you will learn how to write stored functions and triggers.

Exercise 1 – Stored Functions

Stored functions are very similar to stored procedures, with the exception that they must return a value to the statement from which they are called.[1] This means you cannot simply execute a stored function; you must "call" the function from within the SELECT line of a query.

1. Write a stored function that uses the year as the input, and calculate the total cost of venue hire, equipment hire, catering and security in that year.

```
TO_C  TOTALCOST
---- ----------
2016       41061
2017       77716
2018       21094
```

2. Write a stored function that takes event id as its input and returns "Sold Out!" if the corresponding event is sold out, "Get in Quick!" if the event has less than 25% of its tickets left, "Selling Steadily" if the event has between 25% and 50% of its tickets left and "More Promotion Required!" if more than 50% of the tickets are left. You need to write the SQL statements to display the ticket sales status of events since 2017 until now.

```
18 rows selected.
```

[1] Price, J. *Oracle Database 10g SQL*. McGraw-Hill, 2005.

Exercise 2 – Triggers

Triggers are procedures that are run (fired) when a specific condition (defined by the trigger) is met during an `INSERT`, `UPDATE` or `DELETE` on a specified table.

RAISE_APPLICATION_ERROR

We can use RAISE_APPLICATION_ERROR to create our own error messages to handle exceptions/errors. Raise application errors can be used to stop a specified action taking place.

Syntax
RAISE_APPLICATION_ERROR (error_number, error_message);
error_number: a parameter between –20000 and –20999
error_message: The (custom) text associated with the error (less than 512 characters)

Eg: RAISE_APPLICATION_ERROR (-20000, 'Customer already in the system');

1. Write a trigger to raise an error when an attempt is made to insert a promoter twice into the system. Note: the trigger is to fire if an existing promoter is allocated a new id, so attributes such as company name and address should be checked. (Think carefully about whether the trigger should be executed before or after the insert.)

2. Write a trigger that prevents the user allocating an event a venue that does not have sufficient capacity.

3. Write a trigger that will create a back-up of the event name and client id if an event is cancelled (deleted). If security or catering had been requested the trigger should remind the user to cancel these services. To create the primary key use a sequence that starts at 100 and increments by 1. You will need to create the back-up table.