# Advanced Databases

## Triggers – Lab Specification

In this lab you will be working on triggers. You will be submitting parts of this lab exercise to your portfolio.

For these exercises, you will need some resources from Brightspace:

- W6-TableSetup.sql: will create the tables and populate with data;
    - While the tables are similar to those used in Week 5, there are some important changes.

1. **Setup**
   Run the script to create the tables and setup the data.

2. **Using Triggers to enforce data validity**
   The business has a rule stating that when a customer is added, their credit limit cannot be set to more than €500 unless they are a VIP customer in which case their limit cannot exceed €1000.
    - Create a trigger to enforce this rule.
    - The trigger should fire whenever an attempt is made to insert a new row into the customers table and output a suitable error message when the rule is violated.
    - Write appropriate SQL to test that your trigger is working as expected.

3. **Using triggers to prevent updates**
   The business has a rule stating that once a shop has been created, its country cannot be changed.
    - Create a trigger to enforce this rule.
    - The trigger should fire each time an update is attempted on the shop table and output a suitable error message when the rule is violated.
    - Write appropriate SQL to test that your trigger is working as expected.

4. **Using triggers to prevent deletion**
   While it is possible to use referential constraints to enforce integrity, triggers can also be used. It should not be possible to delete a customer if sales have been made to that customer
    - Create a trigger to enforce this rule.
    - The trigger should fire whenever an attempt to delete is attempted on the customer table and output a suitable error message when the rule is violated.
    - Write appropriate SQL to test that your trigger is working as expected.

5. **Using triggers after events**

   The business wants to maintain an audit trail of users and sales they are responsible for. A table, sales_audit, has been created for this purpose (see SQL file for details).

   The business wishes to record for each sale made, the username of the user who executed the SQL.

   - Create a trigger to achieve this.
   - The trigger should fire after each insert into the sales table.
   - Write appropriate SQL to test that your trigger is working as expected.
   - Note:
     - You can retrieve the username from dual e.g. SELECT user FROM dual;

6. **Create an additional trigger to achieve ONE of the following:**
   - **a.** Prevent the insertion of sales between 11pm and midnight (hint SYSTIMESTAMP).
     - Write appropriate SQL to test that your trigger is working as expected.
   - **b.** Ensure that the names of customers are only changed during 11pm and midnight (hint SYSTIMESTAMP).
     - Write appropriate SQL to test that your trigger is working as expected.