

Triggers exercises

The tasks below assume existence of the following data structures:

```
CREATE TABLE Emp_tab (  
Empno NUMBER NOT NULL,  
Ename VARCHAR2(10),  
Job VARCHAR2(9),  
Mgr NUMBER(4),  
Hiredate DATE,  
Sal NUMBER(7,2),  
Comm NUMBER(7,2),  
Deptno NUMBER(2) NOT NULL);
```

```
CREATE TABLE Audit_table (  
Seq NUMBER,  
Time_now DATE,  
Job VARCHAR2(10),  
Proc VARCHAR2(10),  
enum NUMBER);
```

```
CREATE SEQUENCE Audit_seq;
```

```
CREATE TABLE Audit_table_values (  
Seq NUMBER,  
Dept VARCHAR2(10),  
Dept1 NUMBER,  
Dept2 NUMBER);
```

```
CREATE TABLE Inventory (  
Part_Number NUMBER,  
Parts_on_hand NUMBER,  
Description VARCHAR2(50),  
Price NUMBER,  
Quantity_On_Hand NUMBER,  
Reorder_point NUMBER,  
Reorder_Quantity NUMBER);
```

```
CREATE TABLE Pending_orders (  
Part_Number NUMBER,  
Reorder_Quantity NUMBER,  
Curr_date DATE);
```

```
CREATE TABLE Audit_emp (  
Emp_no NUMBER,  
Description VARCHAR2(30),  
Date_inserted DATE);
```

1. Create a row trigger for the *Inventory* table. The trigger should be fired after an *update* event on the column *Parts_on_hand*. If the new value for *Parts_on_hand* is greater than the new value for *Reorder_point*, the trigger should count the number of such rows in the *Pending_orders* table, where the value for *Part_Number* is equal to the inserted part number *Part_Number*. In the case when this value is 0 it should insert record to the *Pending_orders* table with the number *Part_Number*, new value *Reorder_quantity* and current date.
2. Create **one** row trigger for the *Emp_tab* table. The trigger should be fired after *insert*, *update* or *delete*.
 - If *insert* occurs, the trigger should insert into the *Audit_table* table the following information: the next value from the `Audit_seq` sequence, current time, 'Emp_tab', 'Insert' and the number of an employee inserted to the *Emp_tab* table.
 - If *delete* occurs, it should log in the *Audit_table* similar information as above, but the event name should be 'Delete' and the value for the last column should be an identifier of the removed employee.
 - If *update* occurs it should log analogous information as in the case of the *delete* but the event name should be 'Update'. If update concerned columns *Sal* or *Deptno*, it should insert old and new values of the modified columns to the table `Audit_table_values` together with the current `Audit_seq` sequence number and information on the name of the modified column (*Sal* or *Deptno*).
3. Create a row trigger for the *Emp_tab* table. The trigger should be fired after the *insert* event and call a stored procedure that inserts a new record to the *Audit_emp* table containing an id number of the inserted employee, the message 'A new record has been inserted' and the current date.