



Exercise Nineteen

These first two exercises can be done either using DBMS_SQL or Execute Immediate, or a mixture of both methods, the solutions to the exercise will contain both versions.

19.1 Create a procedure called `disable_triggers` to loop through each trigger in your schema and disable each in turn. Hint: Use the metadata table - `user_triggers`.

Save as `c:\course\ex19_1.sql`

19.2 Modify `c:\course\ex19.1.sql` to either enable or disable all triggers according to a parameter passed at execution. Save as `c:\course\ex19_2.sql`

Exercise Nineteen



19.3 Modify the trigger created in c:\course\ex13_2 so that the errors are now written to an audit file named <user>si.err in the 'COURSE' oracle directory.

Save as c:\course\ex19_3.sql

The output should be tested as follows ...

```
insert into stock_information values (1,1,'Testing1',12,25,user,sysdate,null,null)
```

ERROR at line 1:

ORA-20000: Cannot insert a Duplicate Stock Number.

ORA-06512: at "PLSQL1001_USER.SI_BI_TRG", line 96

ORA-04088: error during execution of trigger 'PLSQL1001_USER.SI_BI_TRG'

21-NOV-10 Error inserting a record into Stock_Information: An attempt was made to insert a Duplicate Stock Number.



Exercise Nineteen

19.4 Create a cursor called `get_stock` in the `seercomp` package made up of the following select clause (Use the version from `c:\course\ex12_4.sql`) ...

```
select si_stock_number as stock_number  
  ,si_section_number as section_number  
  ,sd_section_name as section_name  
  ,si_stock_description as stock_description  
  ,si_unit_price as stock_price  
  ,si_no_in_stock as stock_in_hand  
  ,si_unit_price * si_no_in_stock as stock_value  
from section_details  
join stock_information on (si_section_number = sd_section_number);
```

Save as `c:\course\ex19_4.sql`

Exercise Nineteen



19.5 Create a new procedure called `process_stock` which utilises the `get_stock` cursor created in the `seer_package`. Loop through each record using a 'for' loop.

Save as `c:\course\ex19.5.sql`. The output from `process_stock` should be as follows ...

| | | |
|-----|---------------------------------|--------|
| 25 | DOORS AND LOCKS HINGE PACK (LH) | £6.99 |
| 9 | DOORS AND LOCKS YALE LOCK | £9.99 |
| 1 | DOORS AND LOCKS BRASS HANDLE | £12.99 |
| 10 | DOORS AND LOCKS BRASS KNOCKER | £15.99 |
| 999 | DOORS AND LOCKS CHISEL | £12.00 |

Exercise Nineteen



19.6 This exercise will use Bulk Collect, create a new procedure called `process_sections` within it create a series of Associate arrays which match individual columns in the `Section_Details` table. Using bulk collect populate these arrays from the `section_details` table. Using a loop print out the contents of the arrays. Save as `c:\course\ex19_6.sql`

execute process_sections

1 DOORS AND LOCKS HARDWARE

2 BATHROOM FITTINGS HARDWARE

3 BULBS AND BLOOMS GARDENING

4 TOOLS AND MOWERS GARDENING

5 TREES GARDENING

6 CEMENT AND SAND BUILDING

7 HAMMERS AND NAILS HARDWARE

8 IRONMONGRY HARDWARE

9 HEADQUARTERS HQ



Exercise Nineteen

19.7 If there is sufficient time, create a copy of the `section_details` table using the following command ...

```
create table section_copy  
select * from section_details  
where 1 = 2;
```

This will create the structure but not copy the data.

Modify the procedure `process_sections` created in `c:\course\ex19_6.sql` to bulk bind the data from the `section_details` into the `section_copy` table. Save as `c:\course\ex19_7.sql`

Exercise Nineteen



19.7 continued ...

```
select * from section_copy
```

```
SD_SECTION_NUMBER SD_SECTION_NAME          SD_SECTION
-----
          1 DOORS AND LOCKS                HARDWARE
          2 BATHROOM FITTINGS              HARDWARE
          3 BULBS AND BLOOMS                GARDENING
          4 TOOLS AND MOWERS                GARDENING
          5 TREES                           GARDENING
          6 CEMENT AND SAND                 BUILDING
          7 HAMMERS AND NAILS               HARDWARE
          8 IRONMONGRY                      HARDWARE
          9 HEADQUARTERS                    HQ
```

9 rows selected.

Exercise Nineteen



19.8 If there is sufficient time, create a new process called `delete_sections` which will delete from `section_copy` and bulk collect the details of the records deleted, output these details. Save as `c:\course\ex19_7.sql`

execute delete_sections

The following records were deleted ...

DOORS AND LOCKS

BATHROOM FITTINGS

BULBS AND BLOOMS

TOOLS AND MOWERS

TREES

CEMENT AND SAND

HAMMERS AND NAILS

IRONMONGRY

HEADQUARTERS