Tshwane University of Technology

We empower people

YEAR: 2011

EXAMINATION: NOVEMBER MAIN

SUBJECT: DEVELOPMENT SOFTWARE IIB/ INFORMATION SYSTEMS IIIB

SUBJECT CODE: DSO23BT/ISY34AB

QUALIFICATION(S): ND: IT & ND: FIS

PAPER: ONLY

PAPER DESCRIPTION: CLOSED BOOK

DURATION: 2 HOURS 30 MINUTES

SPECIAL REQUIREMENTS

- [x] NONE
- [ ] NON-PROGRAMMABLE POCKET CALCULATOR
- [ ] SCIENTIFIC CALCULATOR
- [ ] COMPUTER ANSWER SHEET
- [ ] GRAPH PAPER
- [ ] DRAWING INSTRUMENTS

OTHER:

INSTRUCTIONS TO CANDIDATES: ANSWER ALL QUESTIONS

TOTAL NUMBER OF PAGES INCLUDING COVER PAGE: 9

TOTAL NUMBER OF ANNEXURES:

EXAMINER: P MUKALA

MODERATOR: SK MOGADI

TOTAL MARKS: 122

FULL MARKS: 120

QUESTION 1 [13]

Write a PL/SQL anonymous block that will display the details about a given project number. If the user provides a project number that does not exist, a suitable error message must be displayed, as demonstrated in the screenshots below.

Example 1:

```
Enter value for project_number: 2

Details about Project Number: 2
****************************************
The project name is: Hardware Support Interface
The client name is: Supreme Data Corporation
They  can be reached on: 5205559821

PL/SQL procedure successfully completed.
```

Example 2:

```
Enter value for project_number: 9

The project does not exist; please provide an existing project number

PL/SQL procedure successfully completed.
```

QUESTION 2 [12]

You are asked to produce information for human resources allocation. Create a PL/SQL anonymous block that will retrieve and display the details about the project with the most hours spent by a consultant. You can assume that there is only one project with this amount of hours spent by a consultant. Refer to the example below. (Do not hardcode).

```
The project that took longest is: Hardware Support Intranet
It took: 245 hours
The consultant was Mr/Mrs.: Paul Courtlandt

PL/SQL procedure successfully completed.
```

QUESTION 3 [15]

Write a PL/SQL anonymous block that will prompt the user to enter a particular evaluation reference number and displays the related information as in the sample output. Make use of a record to store and display the information.

Example 1:

```
Enter value for evaluation_number: 104

On this date<< 29-DEC-02 The evaluator number 104 conducted
an evaluation for the  "Hardware Support Intranet" project and
the score was 75

PL/SQL procedure successfully completed.
```

Example 2:

```
Enter value for evaluation_number: 102

On this date<< 08-JAN-03 The evaluator number 105 conducted
an evaluation for the  "Hardware Support Intranet" project and
the score was 90

PL/SQL procedure successfully completed.
```

QUESTION 4 [20]

Making use of an index by table of records, create an anonymous block that will retrieve all the information there is about projects and display data relevant to the user input. The user is requested to input a letter and the program displays the information related to a project name that ends with that letter. You can assume that there are 7 projects, as indicated in the relevant table. Study the examples below.

Example 1:

```
Enter value for letter: e
Project(s) matching your letter "e" description:
==========================================================
Project => 2,  <<Hardware Support Interface>>  from client number:2
Project => 3,  <<Hardware Support Database>>  from client number:2
Project => 7,  <<Exploration Database>>  from client number:5

PL/SQL procedure successfully completed.
```

Example 2:

```
Enter value for letter: t
Project(s) matching your letter "t" description:
==========================================================
Project => 1,  <<Hardware Support Intranet>>  from client number:2

PL/SQL procedure successfully completed.
```

QUESTION 6 [47]

6.1 Write a procedure **consproc** that will use a cursor and parameterized cursor to produce a master-detail report of consultant information. The report must show the consultant's personal information, the projects that he/she worked on, the roll-on dates of the projects, as well as the hours spent on each of these projects. It must also show the total project hours for each of the consultants. The procedure will take two parameters: one for the start date and one for the end date of the time period for the report. The report must include all projects that had their **roll_on_dates** within the specified time period.

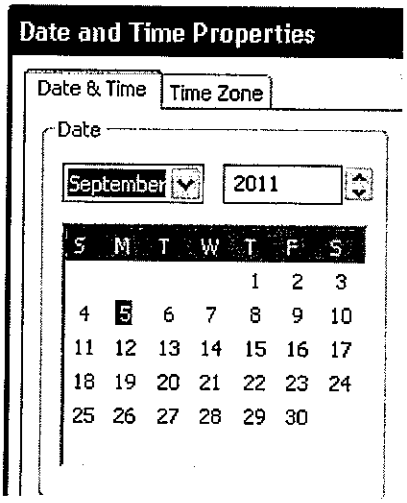
Consultants must be displayed in alphabetical order, as shown. The projects on which they worked must be displayed so that the projects on which they spent most hours are at the top, as shown in the screenshot. Ensure that the fields are properly aligned, as shown, with the correct formats. (28)

```
SQL> execute consproc ('1-SEP-2002','30-SEP-2002')
Consultant information
----------------------
103 Carlson, S.
Project name                    Roll on date        Total hours spent
*******************************************************************************
Hardware Support Database       15-September-2002    125
Internet Advertising            19-September-2002     15
Network Design                  19-September-2002      5

Total hours for this consultant: 145
====================================
Consultant information
----------------------
104 Courtlandt, P.
Project name                    Roll on date        Total hours spent
*******************************************************************************
Network Design                  19-September-2002     10

Total hours for this consultant: 10
===================================
Consultant information
----------------------
105 Park, J.
Project name                    Roll on date        Total hours spent
*******************************************************************************
Internet Advertising            19-September-2002     15

Total hours for this consultant: 15
===================================

PL/SQL procedure successfully completed.
```

```
SQL> execute consproc('1-SEP-2002','35-SEP-2002')

Some error occurred concerning the dates you entered - please re-enter
```

6.3 Write a function **tothours** that will determine the total amount of working hours between two specified dates (including both of these dates). The working hours are determined as follows: If the day is in the week (Monday to Friday), 8 working hours are added. For weekends, no working hours are added. The function takes two parameters: one for the start and one for the end date. Hint: You will have to use a loop to work through all the days. (13)

Example



This example illustrates that the day 1,2 5 and 6 September are weekdays, for which 8 hours are added, whereas the weekend (3 and 4 September) do not have working hours. This gives a total of 32 working hours. Note also that both the 1st and the 6th September are included in the calculation.

```
execute :g_hrs := tothours('1-SEP-2011','6-SEP-2011')

Print g_hrs

   G_HRS

------------

        32
```

6.4 Adjust the stored procedure in 6.1 to use the function **tothours** (defined in 6.3), to display the ratio of the hours spent on all of a consultants projects to the total amount of working hours that were available in the specified time period (i.e. divide the total hours for each consultant by the amount of working hours available during the specified time period). The output of the stored procedure will now appear as shown in the screenshot. (2)

```
SQL> execute consproc ('1-SEP-2002','30-SEP-2002')
Consultant information
----------------------
103 Carlson, S.
Project name                    Roll on date        Total hours spent
*************************************************************************
Hardware Support Database       15-September-2002    125
Internet Advertising            19-September-2002     15
Network Design                  19-September-2002      5

Total hours for this consultant: 145 that is  0.86 of available work hours
=========================================================================
Consultant information
----------------------
104 Courtlandt, P.
Project name                    Roll on date        Total hours spent
*************************************************************************
Network Design                  19-September-2002     10

Total hours for this consultant: 10 that is  0.06 of available work hours
=========================================================================
Consultant information
----------------------
105 Park, J.
Project name                    Roll on date        Total hours spent
*************************************************************************
Internet Advertising            19-September-2002     15

Total hours for this consultant: 15 that is  0.09 of available work hours
=========================================================================

PL/SQL procedure successfully completed.
```

QUESTION 7 [15]

Write a stored procedure **insertskill** that will be used to insert skills in the SE_SKILL table and insert skills in the SE_CONSULTANT_SKILL table at the same time.

The procedure will have one parameter for the skill_description. The procedure must insert the given skill into the SE_SKILL table by calculating a suitable se_skill_id. Once this is done, this skill must be added for all the consultants, **using a cursor.** When the skill is added for each consultant, the certification must be recorded as N.

Example (the added records are shown in bold)

```
SQL> execute insertskill('Quality assurance')

PL/SQL procedure successfully completed.

SQL>SELECT * FROM SE_SKILL;

SE_SKILL_ID SE_SKILL_DESCRIPTION
----------- ----------------------------------------
          1 Visual Basic Programming
          2 COBOL Programming
          3 Java Programming
          4 SE_project Management
          5 Web Application Programming
          6 Oracle Developer Programming
          7 Oracle Database Administration
          8 Windows NT Network Administration
          9 Windows 2000 Network Administration
         10 Quality assurance

10 rows selected.

SQL>SELECT * FROM SE_CONSULTANT_SKILL;
    C_ID       SE_SKILL_ID CERTIFICATION
---------- ----------- ----------------------------------------
       100           1 Y
       100           3 N
       100           6 Y
       101           4 N
       101           5 N
       102           7 Y
       103           1 Y
       103           6 Y
       103           8 Y
       103           9 Y
       104           8 N
       104           9 Y
       105           2 N
       105           3 N
       105           4 Y
       100          10 N
       101          10 N
       102          10 N
       103          10 N
       104          10 N
       105          10 N

21 rows selected.
```

Appendix A Software Database