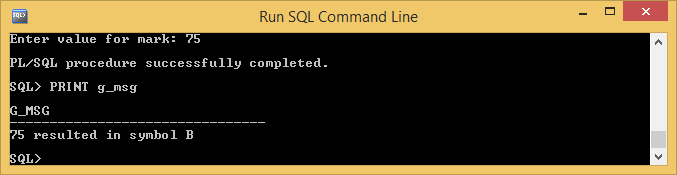
|  |  |
| --- | --- |
|  | FACULTY Of INFORMATION, COMMUNICATION AND TECHNOLOGY  DEPARTMENT OF SOFTWARE ENGINEERING |
| SEMESTER TEST 1 MEMORANDUM  Subject Code : DSO23BT/SFW20BT  Subject Name : Development Software 2B/Software Skills 2B  Examiner(s) : GO Leroke, EH Mathonsi  Moderator : SK Mogapi  Marks : 65 Time : 180 Minutes  Total Pages( Including this one) : 5 | |
| **Rules**   1. Read and understand the question before you answer. Answer all the questions. 2. Your answers should produce the same output as the ones the examiner(s) has given as examples. 3. Complete the information on this cover page before you start answering questions. 4. The question paper is based on the **Summit Sporting Goods database**. Install the database on your computer. 5. Complete required details on the top of the notepad and save it as **YourStudentNUmberST1\_2015**. 6. It’s your responsibility to make it a point that you follow the rules. Empty notepad means 0% mark. | |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Student No : |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | | Surname and Initials : |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | Lecturer’s Name : |  |  | Computer # : |  | | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Questions** | **Question 1** | **Question 2** | **Question 3** | **Question 4** | **Question 5** | **Total** | | **Marks** | **09** | **11** | **14** | **13** | **18** | **65** | | **Allocation** |  |  |  |  |  |  |       % | |

**Question 1 [9]**

Write a PL/SQL block that prompts the user to enter a mark and later display the appropriate symbol to represent that mark. Make use of a use case expression to determine the symbol of the mark according to the table below. Make use of BIND/ HOST variable as shown in the output.

|  |  |
| --- | --- |
| Mark | Symbol |
| 90+ | A+ |
| 80+ | A |
| 70+ | B |
| 60+ | C |
| 50+ | D |
| Default | F |



VARIABLE g\_mark NUMBER½

VARIABLE g\_results VARCHAR2(10) ½

VARIABLE g\_msg VARCHAR2(30) ½

BEGIN

:g\_mark := &mark; ½

:g\_results :=

CASE ½

WHEN :g\_mark >= 90 THEN 'A+'✓

WHEN :g\_mark >= 80 THEN 'A'✓

WHEN :g\_mark >= 70 THEN 'B'✓

WHEN :g\_mark >= 60 THEN 'C'✓

WHEN :g\_mark >= 50 THEN 'D'✓

ELSE 'F'½

END;

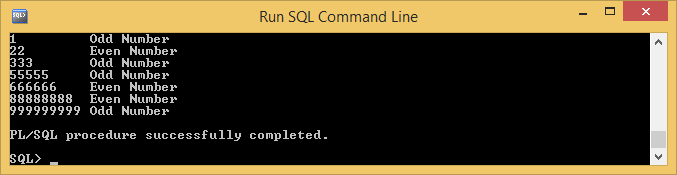
:g\_msg := :g\_mark||' resulted in symbol '||:g\_results; ✓

END;

/

**Question 2 [11]**

Create a PL\SQL block that displays numbers from 1 to 9 excluding numbers 4 and 7, indicate if the number is odd or even. Your output should resemble the one below.



BEGIN

FOR j IN 1..9 LOOP✓

IF j = 4 THEN½ Or IF j=4 OR j=7 THEN NULL;

DBMS\_OUTPUT.PUT\_LINE(' '); ½

ELSIF j = 7 THEN½ 2

DBMS\_OUTPUT.PUT\_LINE(' '); ½

ELSE½

IF MOD(j,2) = 1 THEN✓ Or j MOD 2 = 1

DBMS\_OUTPUT.PUT\_LINE(RPAD(LPAD(' ',j+1,j) ✓,10,' ') ✓||'Odd Number'½);

ELSE

DBMS\_OUTPUT.PUT\_LINE(RPAD(LPAD(' ',j+1,j) ✓,10,' ') ✓||'Even Number'½);

END IF; ½

END IF; ½

END LOOP; ½

END;

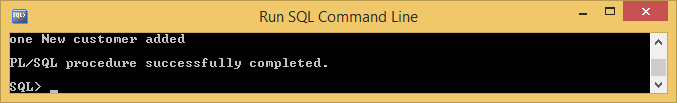
/

**Question 3 [14]**

* 1. Create a PL/SQL block that adds a new customer as detailed in the table below. **(9)**

|  |  |
| --- | --- |
| Field | Data |
| ID | Use sequence named **s\_customer\_id** to populate |
| Name | TUT Sporting |
| Phone | (012) 382 0628 |
| Address | TUT Number 1 |
| City | Pretoria |
| State | Gauteng |
| Country | South Africa |
| Zip Code | 0001 |
| Credit Rating | GOOD |
| Sales Rep id |  |
| Region id | 3 |
| Comments |  |

After successful addition, the following message should be displayed:



BEGIN

INSERT INTO s\_customer ✓

VALUES (s\_customer\_id.NEXTVAL✓,'TUT Sporting'½,0123820628½,'TUT Number 1'½,'Pretoria'½,'Gauteng'½,'South Africa'½,'0001'½,'GOOD'½,NULL½,3½,NULL½);

DBMS\_OUTPUT.PUT\_LINE('one New customer added');

END;

/

* 1. Create a PL/SQL program that will update customer’s information for customer number 216. **(5)**

phone to +26461240555

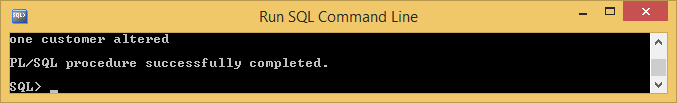
address to 10 Tienie Louw Street

city to Windhoek

state to Khomas Highland

zip\_code to 9000

After successful update, the following message should be displayed:



BEGIN

UPDATE s\_customer✓

SET phone = '+26461240555', ½

address = '10 Tienie Louw Street', ½

city = 'Windhoek', ½

state = 'Khomas Highland', ½

zip\_code = '9000'½

WHERE id = 216; ½

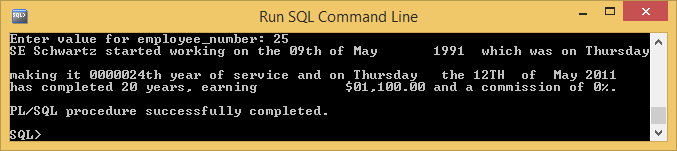
DBMS\_OUTPUT.PUT\_LINE('one customer altered'); ✓

END;

/

**Question 4 [13]**

Create a PL/SQL that declares a record named **emp\_record** using %ROWTYPE. The program must prompt the user to enter employee number then return the initials , surname, hire date, number of years employed. The program must also calculate the exact date that an employee completed 20 years of service, and also determine the salary and commission since the day of week the employee started. Format your output to resemble the output below:



DECLARE

**emp\_record** s\_emp%ROWTYPE; ✓

BEGIN

SELECT \*

INTO emp\_record ✓

FROM s\_emp

WHERE id = &employee\_number; ✓

DBMS\_OUTPUT.PUT\_LINE(

SUBSTR(emp\_record.first\_name,1,1)|| ✓

||' '||emp\_record.last\_name||' started working on the '

||TO\_CHAR(emp\_record.start\_date,'ddth "of" Month YYYY " which was on "Day') ✓

||' making it '||TO\_CHAR(TO\_DATE(ROUND(MONTHS\_BETWEEN(SYSDATE,emp\_record.start\_date)/12,0),'j'),'jth') ✓✓||' year of service and on '||TO\_CHAR(NEXT\_DAY(ADD\_MONTHS(emp\_record.start\_date,240),'THURSDAY'),'Day " the "DDth " of " Mon YYYY ') ✓✓✓

||' has completed 20 years, earning '||TO\_CHAR(emp\_record.salary,'L09,999.00') ✓

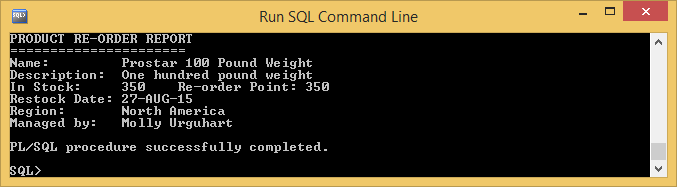
||' and a commission of '||NVL(emp\_record.commission\_pct,0) ✓||'%.'); ✓

END;

/

**Question 5 [18]**

Create a PL/SQL that declares a record named **product\_record.** This record must prompt the user to enter product id, and displays product name and description, the quantity in the inventory and re-order point, region name and the manager’s details for products that have same amount of quantity in and re-order point. Define the product id to 41100. Do not use %ROWTYPE.



DEFINE product\_code = 41100✓

DECLARE

TYPE product\_record\_type IS RECORD✓

(p\_name s\_product.name%TYPE, ½

p\_short\_desc s\_product.short\_desc%TYPE, ½

i\_amount\_in\_stock s\_inventory.amount\_in\_stock%TYPE, ½

i\_restock\_date s\_inventory.restock\_date%TYPE, ½

i\_reorder\_point s\_inventory.reorder\_point%TYPE, ½

r\_name s\_region.name%TYPE, ½

e\_last\_name s\_emp.last\_name%TYPE, ½

e\_first\_name s\_emp.first\_name%TYPE); ½

product\_record product\_record\_type; ✓

BEGIN

SELECT p.name,p.short\_desc,i.amount\_in\_stock,NVL(i.restock\_date,0) ✓,i.reorder\_point,r.name,e.last\_name,e.first\_name

INTO product\_record½

FROM s\_inventory i,s\_product p,s\_warehouse w,s\_region r,s\_emp e✓✓½

WHERE i.product\_id = p.id½

AND i.warehouse\_id = w.id½

AND w.region\_id = r.id½

AND w.manager\_id = e.id½

AND p.id = &product\_code½

AND i.amount\_in\_stock = i.reorder\_point½;

DBMS\_OUTPUT.PUT\_LINE('PRODUCT RE-ORDER REPORT');

DBMS\_OUTPUT.PUT\_LINE('======================');

DBMS\_OUTPUT.PUT\_LINE('Name: '||product\_record.p\_name); ½

DBMS\_OUTPUT.PUT\_LINE('Description: '||product\_record.p\_short\_desc); ½

DBMS\_OUTPUT.PUT\_LINE('In Stock: '||product\_record.i\_amount\_in\_stock||' Re-order Point: '||product\_record.i\_reorder\_point); ✓

DBMS\_OUTPUT.PUT\_LINE('Restock Date: '||product\_record.i\_restock\_date); ½

DBMS\_OUTPUT.PUT\_LINE('Region: '||product\_record.r\_name); ½

DBMS\_OUTPUT.PUT\_LINE('Managed by: '||product\_record.e\_first\_name||' '||product\_record.e\_last\_name); ✓

END;

/