|  |  |
| --- | --- |
|  | FACULTY Of INFORMATION, COMMUNICATION AND TECHNOLOGY  DEPARTMENT OF SOFTWARE ENGINEERING |
| CLASS TEST 2 MEMORANDUM  Subject Code : DSO23BT/SFW20BT  Subject Name : Development Software 2B/Software Skills 2B  Examiner(s) : MC Phiri  Moderator : SK Mogapi  Marks : 50 Time : 120 Minutes  Total Pages( Including this one) : 4 Annexure(s) : 1 | |
| **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 **YourStudentNUmberCT2\_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** | **Total** | | **Marks** | **15** | **20** | **15** | **50** | | **Allocation** |  |  |  |  |       % | |

**QUESTION 1 [15]**

1. Write PL/SQL block that create an explicit cursor named **total\_comm\_cur** to retrieve the total sales of orders generated by a specific salesperson. The cursor must interactively ask the user for the sales\_person\_id. Update the commission of that specific sales\_person by multiplying his/her total by commission\_pct and divide by 12. Your cursor must lock all the record to be updated. (15)

***DECLARE***

***CURSOR total\_comm\_cur IS🗸***

***SELECT sales\_rep\_id, commission\_pct,total🗸***

***FROM s\_ord o, s\_emp e🗸***

***WHERE sales\_rep\_id =&salesperson\_id🗸***

***AND e.id=o.sales\_rep\_id ½***

***FOR UPDATE OF commission NOWAIT; 🗸 for commission as a field or 0 for incorrect field.***

***v\_total\_sales NUMBER(13,2):=0; ½***

***v\_monthly\_sales NUMBER(13,2):=0; ½***

***v\_empno s\_emp.id%TYPE;***

***v\_comm\_rate s\_emp.commission\_pct%TYPE;***

***V\_total s\_ord.total%TYPE;***

***BEGIN***

***OPEN total\_comm\_cur; 🗸***

***LOOP ½***

***FETCH total\_comm\_cur INTO v\_empno,v\_comm\_rate,v\_total; 🗸***

***EXIT WHEN total\_comm\_cur%NOTFOUND; 🗸***

***v\_total\_sales:= v\_total\_sales + v\_total; 🗸***

***END LOOP;***

***v\_monthly\_sales:=(v\_monthly\_sales + v\_total\_sales)/12; 🗸***

***UPDATE s\_emp 🗸***

***SET commission = (v\_monthly\_sales \* (v\_comm\_rate/100)) 🗸***

***WHERE id = v\_empno; 🗸***

***END;***

***/***

**QUESTION 2 [20]**

Write a PL/SQL block that creates an explicit cursor named **customer\_details\_cur** that accept the customer\_id interactively from the user and retrieve the customer’s name and customer\_id.

Create a second cursor named **customer\_order\_cur** that uses a parameter **p\_cust\_id** that accept the value passed to it from the first cursor. It must display all the product the customer has ordered and their quantity, unit prices and subtotal. Then the block must display the grand total which must be equal to the total of all orders made by the customer and must be the same as the value in order table for that order number.

**Enter value for cust\_no:** 203

Customer: Delhi Sports(203) has made the following orders :

===================================================

Order No Product Details Quantity Unit Price Sub Total

======== ========================= ======== ============ ===========

99 Black Hawk Knee Pads(20510) 18 R9.00 R162.00

99 Black Hawk Elbow Pads(20512) 25 R8.00 R200.00

99 Griffey Glove(50417) 53 R80.00 R4,240.00

99 Cabrera Bat(50530) 69 R45.00 R3,105.00

The gran total paid is **R7,707.00**

***/\* CHR(9) is a horizontal tab \*/***

***DECLARE***

***v\_customer\_id s\_customer.id%TYPE:=&cust\_no; ½***

***v\_cust\_name s\_customer.name%TYPE;***

***v\_order\_id s\_ord.id%TYPE;***

***v\_prod\_id s\_item.product\_id%TYPE; ½ for next three***

***v\_prod\_name s\_product.name%TYPE;***

***v\_qty s\_item.quantity%TYPE;***

***v\_unit\_price s\_item.price%TYPE; ½ for next three***

***v\_sub\_total NUMBER(12,2):=0;***

***V\_grand\_total NUMBER(14,2):=0;***

***CURSOR customer\_details\_cur IS ½***

***SELECT id,name ½***

***FROM s\_customer ½***

***WHERE id=v\_customer\_id; ½***

***CURSOR customer\_orders\_cur(p\_cust\_id NUMBER) IS 🗸***

***SELECT o.id,i.product\_id,p.name,i.quantity,i.price, i.quantity \* i.price sub\_total 🗸***

***FROM s\_ord o,s\_item i,S\_product p ½½½***

***WHERE o.customer\_id=p\_cust\_id ½***

***AND i.ord\_id=o.id ½***

***AND p.id=i.product\_id; ½***

***BEGIN***

***OPEN customer\_details\_cur; ½***

***LOOP***

***FETCH customer\_details\_cur ½ INTO v\_customer\_id,v\_cust\_name; ½***

***EXIT WHEN customer\_details\_cur%NOTFOUND; ½***

***DBMS\_OUTPUT.PUT\_LINE('Customer : '||v\_cust\_name||'('||v\_customer\_id||') has made the following orders :'); 🗸***

***DBMS\_OUTPUT.PUT\_LINE('======================================================================='); ½***

***END LOOP;***

***OPEN customer\_orders\_cur(v\_customer\_ID); ½½***

***DBMS\_OUTPUT.PUT\_LINE('Order No'||CHR(9)||'Product Details'||CHR(9)||'Quantity'||CHR(9)||'Unit***

***Price'||CHR(9)||'Sub Total'); ½½***

***DBMS\_OUTPUT.PUT\_LINE('========'||CHR(9)||'==============='||CHR(9)||'========'||CHR(9)||'=========='||CHR(9)|***

***|'========='); ½***

***LOOP ½***

***FETCH customer\_orders\_cur INTO v\_order\_id,v\_prod\_id,v\_prod\_name,v\_qty,v\_unit\_price,v\_sub\_total; 🗸***

***EXIT WHEN customer\_orders\_cur%NOTFOUND; ½***

***DBMS\_OUTPUT.PUT\_LINE(v\_order\_id||CHR(9)||v\_prod\_name||'('||v\_prod\_id||')'||CHR(9)||v\_qty||***

***TO\_CHAR(v\_unit\_price,'fmL99,999,999.00')||CHR(9)||TO\_CHAR(v\_sub\_total,'fmL999,999,999.00')); 🗸🗸***

***v\_grand\_total:=v\_grand\_total + v\_sub\_total; ½***

***END LOOP;***

***DBMS\_OUTPUT.PUT\_LINE('The gran total paid is '||TO\_CHAR(v\_grand\_total,'fmL999,999,999.00')); 🗸***

***CLOSE customer\_details\_cur; ½***

***CLOSE customer\_orders\_cur; ½***

***END;***

***/***

**QUESTION 3 [15]**

The PL/SQL block that will extract information from the database when the user interactively enters customer no. For a customer who has a single order display his/her information like the one below. All other queries create error trapping as per table below:

**Enter value for customer\_no:** 202

OJ Atheletics(202) ordered 7 units of 40421-Alexeyer Pro Lifting Bar at R85.00 per unit and the subtotal is R595.00

|  |  |  |
| --- | --- | --- |
| **Type of error** | **Message** | **Type of exception** |
| Unspecified error | Unknown error | Oracle pre-defined |
| Returns more than one rows | Customer name||’ ‘|| has ordered more than one product. | Oracle non pre-defined(ORA-1422) |
| Returns no rows | Customer name||’ ‘|| has no orders. | User-define |

**Example 1**

**Enter value for customer\_no:** 201

Unisports(201) has ordered more than one product.

**Example 2**

**Enter value for customer\_no**: 200

Unknown error!!!

***DECLARE***

***v\_prod\_id s\_product.id%TYPE;***

***v\_prod\_name s\_product.name%TYPE;***

***v\_cust\_name s\_customer.name%TYPE;***

***v\_cust\_id s\_customer.id%TYPE:=&customer\_no; ½***

***v\_unit\_price s\_item.price%TYPE;***

***v\_quantity s\_item.quantity%TYPE;***

***v\_sub\_total NUMBER(12,2); ½ for the rest***

***e\_no\_orders EXCEPTION; ½***

***e\_more\_orders EXCEPTION; ½***

***PRAGMA EXCEPTION\_INIT (e\_more\_orders, -1422); 🗸***

***BEGIN***

***SELECT p.id,p.name,i.price,i.quantity,c.name,c.id,(i.price\*i.quantity) ½ for calculation and ½ for the rest***

***INTO v\_prod\_id,v\_prod\_name,v\_unit\_price,v\_quantity,v\_cust\_name,v\_cust\_id,v\_sub\_totl 🗸***

***FROM s\_ord o,s\_item i,s\_product p,s\_customer c 🗸🗸***

***WHERE c.id=v\_cust\_id ½***

***AND o.customer\_id=c.id ½***

***AND i.ord\_id=o.id ½***

***AND p.id=i.product\_id; ½***

***DBMS\_OUTPUT.PUT\_LINE(v\_cust\_name||'('||v\_cust\_id||') ordered '||v\_quantity||' units of '||v\_prod\_id||***

***'-'||v\_prod\_name||' at '||TO\_CHAR(v\_unit\_price,'fmL999,999.00')||' per unit and the subtotal is '||***

***TO\_CHAR(v\_sub\_total,'fmL999,999.00')); ½ for 9 formats ½ for currency ½ for the rest***

***IF SQL%NOTFOUND THEN ½***

***RAISE e\_no\_orders; ½***

***END IF;***

***EXCEPTION***

***WHEN e\_more\_orders THEN ½***

***DBMS\_OUTPUT.PUT\_LINE(v\_cust\_name||'('||v\_cust\_id||') has ordered more than one product.'); 🗸***

***WHEN e\_no\_orders THEN ½***

***DBMS\_OUTPUT.PUT\_LINE(v\_cust\_name||'('||v\_cust\_id||') has no orders.'); 🗸***

***WHEN OTHERS THEN ½***

***DBMS\_OUTPUT.PUT\_LINE('Unknown error!!!'); 🗸***

***END;***

***/***

**SUMMIT SPORTING GOODS DATABASE STRUCTURE**

