|  |  |
| --- | --- |
|  | DEPARTMENT COMPUTER SCIENCE  DATA SCIENCE SECTION |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY | | | | | | | | | | | | | | | | Examiner: Mrs BI Ariyo  Moderator: Mr L Mathi  grey_logo_portrait**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature** | **Development Software 2B**  DSO23BT | | | | | | | | | | | | | | | *ASSIGNMENT 2*  MEMO  **Due Date 08 JULY 2021**  **Full Marks: 71** | | | | | | **Class list Number:** | | | | | | | | | **Student number:** | | | | | | | | | | | | | | |  |  |  |  |  | | |  |  | |  | |  | | Surname: | | | | | Initials: | | | | **/71** | | **%** | | |   **Use the Kaleidoscope Database.**  **INSTRUCTIONS ON HOW TO WRITE THE TEST ON ELECTRONIC CAMPUS (EC):**   1. Create a folder on **D:\ Drive** and rename the folder your **student number**. 2. Open a text editor (notepad++ is recommended) or MS Word and save the text file in your student number folder. Note: All Questions must be answered on EC. 3. Open the **browser**. 4. Type <http://ec.tut.ac.za> in the address field of the browser and press enter. 5. Click **Login.** 6. Type in student number for user name and if you are using EC for the first time you must use your student number as your password. 7. Alternatively, you must enter your password. (Tut4life Credentials) 8. Click **Login** **button**. 9. Click [**Continue**] 10. Select [**Student**] 11. Select [**Tests**] 12. Select [**Webtest 4**] 13. Type or Select **DSO23BT** into the first text field. 14. Type or Select **20211** into the second text field. 15. Type or Select **R** into the third text field (The “**R**” must be a capital letter). 16. Click "**Next**" 17. Click on the "**Write**" button 18. Click on "**[Start]**" and you will be in the test to submit your answers 19. Click on **[Section]** on the test to submit each Questions, and click [Go] to submit your answer- Copy your answer and paste it in the space provided 20. Click **[Test]** to move to the next or previous Questions 21. Under each question click edit to update your answer 22. For Question 1 to 4 paste your code into space provided 23. Softcopy of the script and database structure can be downloaded under from bright space in the content folder. | | |

**Question 1 [13]**

1. Create a PLSQL block that prompts the user to input his/her full name and birthdate in the format DD-MON-YYYY. The program must display the user’s Initial, Surname, the total number of characters in the first name, the username and password. The username is derived from the user’s full name and birth year. The password is derived from the first 3 characters of the surname, followed by the @ sign, followed by the initials and followed by the digits 02. Your output should be in the format below:

Text

Description automatically generated

**DECLARE**

**fname VARCHAR2(10) √:= '&Enter\_firstname'; √**

**lname VARCHAR2(10) √ := '&Enter\_lastname'; √**

**bdate DATE √:= '&birth\_date'; √**

**totalchr NUMBER(2); √**

**userid VARCHAR2(20); √**

**pword VARCHAR2(15); √**

**BEGIN**

**SELECT lname, SUBSTR(fname, 1,1) √, LENGTH(fname) √, fname||lname√||SUBSTR(bdate, -2, 2) √, SUBSTR(lname,1, 3)||'@'||√SUBSTR(fname, 1,1)||'02' √**

**INTO lname, fname, totalchr, userid, pword 🗸🗸√**

**FROM dual; √**

**DBMS\_OUTPUT.PUT\_LINE('My name is '|| fname √||' '|| lname ||'√ and my firstname has '||totalchr||√' characters.');**

**DBMS\_OUTPUT.PUT\_LINE('Username: '||userid); √**

**DBMS\_OUTPUT.PUT\_LINE('Password: '||LOWER(pword)); √**

**END;**

**/**

**QUESTION 2 [17]**

## Create a PL/SQL block of record that will prompt the user to enter an input an employee number and then displays the relevant information about the employee; refer to the sample output given below.

Example output:

Text

Description automatically generated

**DECLARE**

**TYPE book\_rec\_TYPE is RECORD √ √**

**(v\_title bk\_books.title%TYPE, √**

**v\_lname bk\_author.lname%TYPE, √**

**v\_fname bk\_author.fname%TYPE, √**

**v\_name bk\_publisher.name%TYPE, √**

**v\_pubdate bk\_books.pubdate%TYPE, √**

**v\_cost bk\_costs.retail%TYPE); √**

**book\_rec book\_rec\_type; √**

**v\_isbn bk\_books.isbn%TYPE √:=’&isbn\_no’; √**

**BEGIN**

**SELECT title, lname, substr√ (fname,1,1) √, name, pubdate, retail√√√**

**INTO book\_rec√**

**FROM BK\_AUTHOR a √, BK\_BOOKS b√, BK\_COSTS c√, BK\_PUBLISHER p√**

**WHERE a.authorid = b.authorid√**

**AND b.pubid = p.pubid√**

**AND b.isbn = c.isbn√**

**AND b.isbn = v\_isbn; √**

**DBMS\_OUTPUT.PUT\_LINE('The book titled '||book\_rec.v\_title√||'('||v\_isbn√||') By '||book\_rec.v\_lname√||' '||book\_rec.v\_fname√**

**||' was published by '||book\_rec.v\_name√||' on the '||TO\_CHAR√ (book\_rec.v\_pubdate,'fmDdth "of" Month, YYYY') √||', and is being sold for '||TO\_CHAR√ (book\_rec.v\_cost, 'fmL9999.99') √||' per unit');**

**END;**

**/**

**Question 3 [19]**

Kaleidoscope Bookshop need to know, from time to time, how many copies of a certain book has been sold.

* Declare a table of type NUMBER(4).
* Declare a cursor that receives one parameter (the ISBN number as supplied by the user). The cursor must retrieve all records from the BK\_ORDERITEMS table with this ISBN number.
* Using a BASIC LOOP, read through the cursor, storing the QUANTITY sold in consecutive elements of the table.
* Use a counter to determine the number of items stored in the table.
* Then use a FOR LOOP to read through the table, finding the sum of all the elements. Display the sum.
* Output Format:

A screenshot of a computer

Description automatically generated with low confidence

**DECLARE**

**TYPE i\_table\_type √is TABLE OF BK\_ORDERITEMS.QUANTITY%TYPE) √√**

**(or TYPE i\_table\_type is TABLE OF NUMBERS(4))**

**INDEX BY BINARY\_INTEGER; √√**

**isbn\_table i\_table\_type; √√**

**CURSOR cur\_items√ (ISBNno bk\_orderitems.isbn%TYPE) √ √**

**IS√**

**SELECT \* √**

**FROM bk\_orderitems√**

**WHERE isbn = ISBNno; √**

**isbn\_rec bk\_orderitems%ROWTYPE; √√**

**counter BINARY\_INTEGER ; √-- NUMBER datatype can also be used**

**total NUMBER(4) ; √**

**BEGIN**

**OPEN cur\_items(‘&isbn’); √√**

**LOOP√**

**FETCH cur\_items √**

**INTO isbn\_rec; √**

**EXIT WHEN√ cur\_items%NOTFOUND; √√**

**counter := counter + 1; √**

**isbn\_table(counter) := isbn\_rec.QUANTITY; √√**

**END LOOP; √**

**FOR cnt IN 1..counter √√**

**LOOP√**

**Total := Total + isbn\_table(cnt); √√**

**END LOOP; √**

**DBMS\_OUTPUT.PUT\_LINE ('We sold: '||total||' of the book'); √√**

**END;**

**/**

**Question 4 [22]**

Write a PL/SQL block to retrieve books from the BK\_BOOKS table based on a PUBID supplied interactively at run time by the user. Use appropriate non-predefined exceptions (using PRAGMA EXCEPTION\_INIT) as defined in the table below to handle the errors and display the message as indicated. otherwise display the book title, the author’s lastname and first name separated by commas, the publisher name and the book category.

|  |  |
| --- | --- |
| **Type of error** | **Message** |
| Unspecified error | Unknown error |
| Returns no books | We do not have any books from that publisher |
| Returns more than one rows | More than one book from that publisher |

**DECLARE**

**v\_title bk\_books.title%TYPE; √**

**v\_fname bk\_author.fname%TYPE; √**

**v\_lname bk\_author.lname%TYPE; √**

**v\_name bk\_publisher.name%TYPE; √**

**v\_category bk\_books.category%TYPE; √**

**v\_pubid bk\_books.pubid%TYPE √:= &publisher\_id; √**

**v\_more\_book exception; √**

**PRAGMA EXCEPTION\_INIT√**

**(v\_more\_book, -1422); √**

**v\_no\_book exception; √**

**PRAGMA EXCEPTION\_INIT √**

**(v\_no\_book, -1420); √**

**BEGIN**

**SELECT title, fname, lname, name, category √√√**

**INTO v\_title, v\_fname, v\_lname, v\_name, v\_category √√√**

**FROM bk\_books, bk\_author, bk\_publisher √√√**

**WHERE bk\_books.pubid =bk\_publisher.pubid√**

**AND bk\_books.authorid = bk\_author.authorid √**

**AND bk\_books.pubid = v\_pubid; √**

**DBMS\_OUTPUT.PUT\_LINE(v\_title ||'written by '||v\_fname||'-'**

**||v\_lname||' publish by '||v\_name**

**||' under '|| v\_category); √√√√√**

**IF SQL%NOTFOUND THEN √√**

**raise v\_no\_book; √**

**ELSIF SQL%ROWCOUNT > 1 THEN √√**

**raise v\_more\_book; √**

**END IF; √**

**EXCEPTION √**

**WHEN v\_no\_book THEN √**

**DBMS\_OUTPUT.PUT\_LINE ('No books from this publisher'); √**

**WHEN v\_more\_book THEN√**

**DBMS\_OUTPUT.PUT\_LINE ('More book from the publisher'); √**

**WHEN OTHERS THEN √**

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

**END;**

**/**