Tshwane University of Technology
We empower people

Tshwane University of Technology

JUNE 2013 MAIN EXAMINATION

SUBJECT: DEVELOPMENT SOFTWARE IIB

SUBJECT CODE: DSO23BT

PAPER NO.: PAPER 1

PAPER DESCRIPTION: CLOSE BOOK

DURATION: 4 HOURS

SPECIAL REQUIREMENTS:	(✓)
None	
Non-programmable pocket calculators	
Scientific calculators	
Drawing instruments	
Other	

INSTRUCTIONS TO CANDIDATES: ANSWER ALL QUESTIONS

NUMBER OF PAGES: 6 (Including this page)
ANNEXURES: 1

QUALIFICATION(S):

N DIP: SOFTWARE DEVELOPMENT	

EXAMINER(S): Mr K MOGAPI / Mr MC PHIRI

MODERATOR: Mr TE MADZUNYE

TOTAL MARKS: 105
FULL MARKS: 100

Tshwane University of Technology
We empower people

INSTRUCTIONS TO CANDIDATES

This paper is a practical paper. The students are required to use a note pad that must be uploaded into a folder which is saved as **student number**. Send the folder to the lecturer's computer or upload it to **ec**. Name the notepad as **DSO23AT_MAIN_JU13**. Write your details at beginning of your note pad and list your questions appropriately. The database to be used is **Kaleidoscope** and will be distributed to the students' computer for them to upload it to their computer's hard drive. Answer ALL the questions.

REQUIREMENTS: Never tear any page from this question paper. Complete this cover page before you start answering the questions.

Time	:	**240 Minutes/ 4 Hours**
Total Marks	:	**105**
Final Marks	:	**100**
Total Pages	:	6 including this page
Annexure	:	**1**
Examiner	:	Mr M.C PHIRI
	:	Mr K. MOGAPI
Moderator	:	Mr T.E MADZUNYE

Sign below to confirm that you understand the instructions above.

SIGNATURE

Tshwane University of Technology

DEPARTMENT Of SOFTWARE ENGINEERING

JUNE 2013 MAIN EXAMINATION

SUBJECT CODE : DSO23BT

SUBJECT NAME : DEVELOPMENT SOFTWARE IIB

COURSE NAME : SOFTWARE ENGINEERING

LECTURER'S NAME	
COMPUTER #	
VENUE	

STUDENT NUMBER								

SURNAME	INITIALS

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total
16	08	07	19	16	12	27	105
							%

Question 1 [16]

Create a PL/SQL programming block that will prompt the user to enter a date in the format as indicated in the example below. Use a **CASE**-statement and convert the month number into the month name as seen in the example. Make provision for a month code that was entered incorrectly.

Example:

```
Please enter a date in the format DD/MM/YYYY: 23/09/1998

The date you entered is: 23 September 1998
```

Question 2 [08]

Write a block of code that receives a user's savings amount and calculates the total savings, making use of **compound** interest (interest on interest), at the end of 12 months. Make use of a BASIC loop to calculate the savings. The interest rate is 15%. Display the new savings using an appropriate message.

Example input:

```
Current savings: 100.00
```

Example output:

```
Your savings at the end of 12 months will be: R465.24
```

Question 3 [07]

Create a PL/SQL block that will prompt the user to input a ISBN number and then displays the relevant book's title and author's first and last name. Refer to the sample output given below.

Example output:

```
ISBN: 8843172113
DATABASE IMPLEMENTATION, WRITTEN BY TINA PETERSON
```

Question 4 [19]

Create a PL/SQL block that declares a record based on the following fields. It prompts the user to enter an order number and later prints the notification invoice slip as indicated in the example below.

Example:

Please enter value for order_No.: 1007

Giana, Tammy From **AUSTIN(TX78710)6** item(s)shipped since **04-April,2003**

Total cost: R347.25
Vat@14%:R48.62
Amount due:R395.87

Please enter value for order_No.: 1002

Mcgovern, Reese From **CHICAGO(IL60606)1** item(s)shipped since **01-April,2004**

Total cost: R55.95
Vat@14%:R7.83
Amount due:R63.78

Question 5 [16]

Create a PL/SQL block uses a cursor that accept an ISBN as parameter and display the information of a book aligned to it. You must display the ISBN, book title, author name, published date, unit price, total units on order and total price of these quantities on order. Format your price using local currency. You've been given to output examples:

Example 1
ISBN **8843172113** is for book **Database Implementation** which was written by **Tina Peterson** and published on the **4/6/1999** at a unit cost of **R55.95** per copy and has **7** quantities on order at a total of **R391.65**

Example 2
ISBN **3957136468** is for book **Holy Grail Of Oracle** which was written by **James Austin** and published on the **31/12/2001** at a unit cost of **R75.95** per copy and has **3** quantities on order at a total of **R227.85**

Question 6 [12]

Create a PL/SQL block that ask for the authored and searches for book he/she has written and if a single book is found display its details others raise exceptions if the following is encountered:

> If no book is found display the message **There are no books for this author**, otherwise if more than one book was retrieved for a specific author display the message **The author has written more than one book** or if none of the above is met display the message Unknown error.

Your output must resemble the one below. Your input must accept the value in any case structure.

Example 1:

```
Enter value for Authorid: k100
Author Name        Book Title                          Publisher Date     Book Price
===============    =========================          ============      ==========
Tamara, Kzochsky   Building A Car With Toothpicks 1    8th March 2002         R59.95
```

Example 2:

```
Enter value for authorid: s100
The author has written more than one books
```

Example 3:

```
Enter value for authorid: m100
The are no books for this author
```

Question 7 [27]

7.1. Create a function **calculate_new_price** that receive a category parameter, calculate and return new prices by using the following percentage of retail price: (09)

Category	Percentage Increment
COMPUTER	7.5%
CHILDREN	2%
FAMILY LIFE	4%
COOKING	5%
BUSINESS	2.5%

7.2. Create a procedure **update_price** which receive a category parameter and uses a **update_price_cursor** cursor which must call the function created in 7.1 to use its values to update the retail of books. If the query returns no books information use and exception **no_cat(-20201)** to display the message **"This category does not have book price"** and if it returns more books per category use the **execption too_many(-01422)** with the message **"This category has more than one book price"**. For a category that return a single book price add the new_price calculated in the function to the retail and update the record and display a message similar to the one in **example 3**. (18)

Example 1

```
SQL> exec update_price('LAPTOPS');
PL/SQL procedure successfully completed.
```

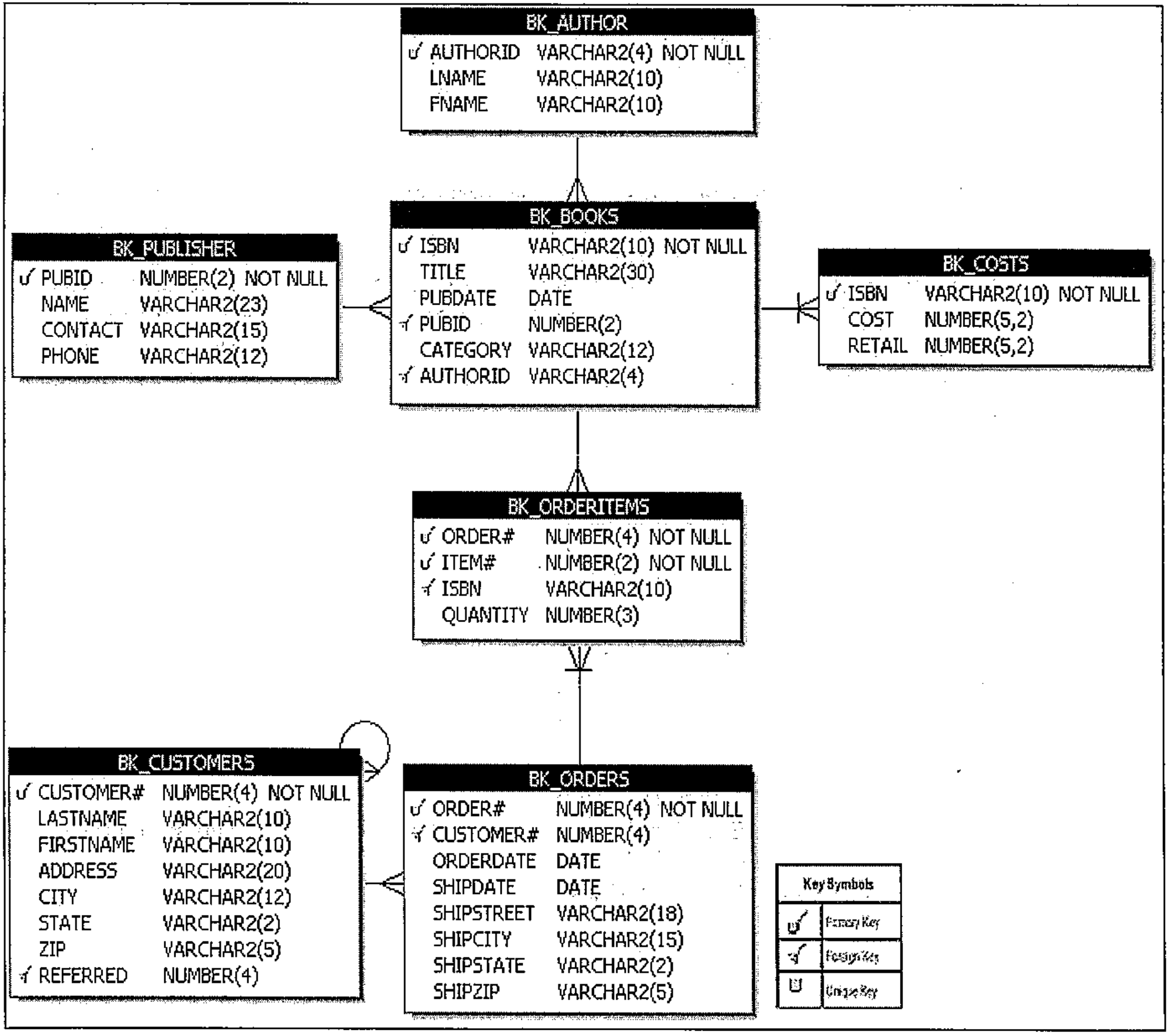
Example 2

```
SQL> exec update_price('COMPUTER');
This category has more than one book price
PL/SQL procedure successfully completed.
```

Example 3

```
SQL> exec update_price('BUSINESS');
HOW TO MANAGE THE MANAGER is written by WILLIAM,WHITE  with selling price of
R33.57 and now it will sell for R34.41
PL/SQL procedure successfully completed.
```

KALEIDOSCOPE DATABASE STRUCTURE


BK_AUTHOR
AUTHORID VARCHAR2(4) NOT NULL
LNAME VARCHAR2(10)
FNAME VARCHAR2(10)
BK_PUBLISHER
PUBID NUMBER(2) NOT NULL
NAME VARCHAR2(23)
CONTACT VARCHAR2(15)
PHONE VARCHAR2(12)
BK_BOOKS
ISBN VARCHAR2(10) NOT NULL
TITLE VARCHAR2(30)
PUBDATE DATE
PUBID NUMBER(2)
CATEGORY VARCHAR2(12)
AUTHORID VARCHAR2(4)
BK_COSTS
ISBN VARCHAR2(10) NOT NULL
COST NUMBER(5,2)
RETAIL NUMBER(5,2)
BK_ORDERITEMS
ORDER# NUMBER(4) NOT NULL
ITEM# NUMBER(2) NOT NULL
ISBN VARCHAR2(10)
QUANTITY NUMBER(3)
BK_CUSTOMERS
CUSTOMER# NUMBER(4) NOT NULL
LASTNAME VARCHAR2(10)
FIRSTNAME VARCHAR2(10)
ADDRESS VARCHAR2(20)
CITY VARCHAR2(12)
STATE VARCHAR2(2)
ZIP VARCHAR2(5)
REFERRED NUMBER(4)
BK_ORDERS
ORDER# NUMBER(4) NOT NULL
CUSTOMER# NUMBER(4)
ORDERDATE DATE
SHIPDATE DATE
SHIPSTREET VARCHAR2(18)
SHIPCITY VARCHAR2(15)
SHIPSTATE VARCHAR2(2)
SHIPZIP VARCHAR2(5)
Key Symbols
Primary Key
Foreign Key
Unique Key