UNIVERSITY OF SWAZILAND

RE-SIT EXAMINATION PAPER 2017/8

TITLE OF PAPER	:	STATISTICAL DATA PROCESSING
COURSE CODE	:	STA206
TIME ALLOWED	:	TWO (2) HOURS
REQUIREMENTS	:	CALCULATOR
INSTRUCTIONS	:	ANSWER ANY THREE (3) QUESTIONS.

Question 1 [20 marks, 3+3+6+4+4]

(a) List with an explanation the various types of database users

(b) List aggregate functions supported by SQL

(c) With an aid of an example explain the Entity-Relationship model

(d) A company wants to move its current file-based system to a database system. In many ways, this can be seen as a good decision. Identify and describe four disadvantages in adopting a database approach.

(e) Database Management Systems provide the following services:

- Concurrent Control
- Recovery
- Authentication
- Integrity

Briefly describe each of the above services.

Question 2 [20 marks, 12+8]

(a) What is 'data editing' and how can it be carried out effectively? Describe the data editing process, illustrating your answer with examples.

(b) Explaining what is meant by the term 'missing data', how it occurs and how can it be dealt with. Illustrate your answer with examples.

Question 3 [20 marks, 2+2+2+2+2+3+2+3+1+1+1+1]

(a) We have three data sequences x_i, y_i and z_i of equal length. These observations are in a file called `exam1.dat`. The first row of this file consists of the first observations x_1, y_1 and z_1, the second row consists of the second observations and so on. Give the R commands to do the following:

(i) Create a data frame called `examinfo` from the `exam1.dat` data file. Name the variables in this data frame x, y and z. Make the variables visible from the command line.

(ii) Calculate the variance for x_i and the covariance between x_i and y_i.

(iii) Assuming there is a unique maximum, print the name of the variable ("x" or "y") that contains the largest value in x and y.

(iv) Plot the points (x_i, y_i) and (x_i, z_i) on the same graph but in different colours.

(v) Output the values of x_i that are greater than y_i but less than or equal to z_i

(vi) Output the index values i for which $x_i = y_i$ or $x_i = z_i$ but $y_i \neq z_i$.

(b) Consider the following specification for an R function:

- takes two vectors (not necessarily the same length) as inputs,

- prints out the relationship (equal to, less than, greater than) between the i^{th} element of the first vector and the i^{th} element of the second vector.

The function func1(...) below represents my first attempt. This function does not meet the specification.

```
> func1 <- function(x1, x2)
+ { lng <- min(length(x1),length(x2))
+ for (i in 1:lng)
+ { cat(i, ": ")
if (x1[i]==x2[i]) cat("EQ. ")
+ if (x1[i]<x2[i]) cat("LT. ")
+ else cat("GT. ")
+ }
+ }
```

[Note: the cat() function simply concatenates its arguments and prints them to screen.] Given the definition of func1(...) above, what is the output from the following commands.

(i) func1(c(2,5,-1,0),3:1)

(ii) func1(1:3, 3:1)

(c) We type the following in R:

```
> theta <- c(8, 6, 4, 2)
> rho <- c(0, 1)
> delta <- c(TRUE,TRUE,FALSE,TRUE,FALSE)
> phi <- seq(from=0, to=8, length=5)
```

Given the assignments above, what is the output of the following commands?

(i) > theta[1:3]

(ii) > theta[-2]

(iii) > theta-rho

(iv) > 3-theta/seq(from=4, to=1)

Question 4 [20 marks, 8+6+6]

Refer to the following tables for this question.

Table 1: transactions

TRANSACTIONID	ACCOUNT_ID	TRANSACTION_DATE	AMOUNT
7659897	93008	12/4/2017	3.67
7659898	93008	12/4/2017	12.99
7743433	93008	13/4/2017	-7.99
7935320	331449	13/4/2017	-14.76
8756571	93008	13/4/2017	-5.99

Table 2: accounts

ACCOUNT_ID	SORT_CODE	ACCOUNT_TYPE	BALANCE
93008	30-54-87	Direct Debit	362.74
331449	31-12-54	Credit	320.26
57746	30-54-87	On-Line Saver	1295.60
16227	12-32-18	Direct Debit	-550.93

(a) List the results of running both the following queries (Query A and Query B) and then describe in a few sentences how these results are produced.

- Query A:

```
SELECT COUNT(*), account_type
FROM accounts
WHERE balance < 4000
GROUP BY account_type
HAVING COUNT(*) > 1;
```

- Query B:

```
SELECT SUM(AMOUNT), t.account_id, transaction_date
FROM transactions t
WHERE t.account_id IN (SELECT a.account_id
FROM accounts a
WHERE account_type <> 'On-Line Saver')
GROUP BY t.account_id, transaction_date
ORDER BY SUM(amount) ASC;
```

(b) Write an SQL query that produces the same output as query B, but instead uses an `INNER JOIN` operator.

For guidance, the syntax of an `INNER JOIN` operator is:-

`INNER JOIN <tablea name> ON <tablea.columna> = <tableb.columna>`

where `columna` is the matching column in `tablea` and `tableb`

(c) Explain the differences between `LEFT` and `RIGHT OUTER JOIN` and an `INNER JOIN`.

Illustrate your answer by showing how replacing an `INNER JOIN` operator with either a `LEFT` or `RIGHT OUTER JOIN` operator can affect the output your answer in part b) above.

Hint : You must choose between either a RIGHT or LEFT OUTER JOIN to illustrate the different output produced compared with using an INNER JOIN