School of Computing and Information Technology

Student to complete:

Family name	
Other names	
Student number	
Table number	

CSCI235
Database Systems

Final Examination Paper
Session 4 2021
(1 December 2021)

Exam duration	3 hours and 40 minutes
Weighting	40% of the subject assessment
Marks available	40 marks
Items permitted by examiner	Text-book, Lecture slides, and Tutorial notes
Directions to students	4 questions to be answered. Marks for each question are shown beside the question. All answers must be written in the answer booklet provided.

This examination is a take-it-home examination to be done on-line on the date of examination.

Version 2.0

Question 3 - (Total 10 marks)
PL/SQL

Time allocated: 45 minutes
Start time: 4:00 pm SGT
End time: 4:45 pm SGT
Submission time start: 4:40 pm SGT
Submission time end: 4:50 pm SGT

Consider a relational database that consists of the relational tables created by the following CREATE TABLE statements:

```
CREATE TABLE CUSTOMER(
C_CUSTKEY           NUMBER(12)      NOT NULL,
C_NAME              VARCHAR(25)     NOT NULL,
C_ADDRESS           VARCHAR(40)     NOT NULL,
C_NATIONKEY         NUMBER(12)      NOT NULL,
C_PHONE             CHAR(15)        NOT NULL,
C_ACCTBAL           NUMBER(12,2)    NOT NULL,
C_MKTSEGMENT        CHAR(10)        NOT NULL,
C_COMMENT           VARCHAR(117)    NOT NULL,
C_TOTORDERS         NUMBER(5),
      CONSTRAINT CUSTOMER_PKEY PRIMARY KEY(C_CUSTKEY),
      CONSTRAINT CUSTOMER_CHECK1 CHECK(C_CUSTKEY >= 0) );

CREATE TABLE ORDERS(
O_ORDERKEY          NUMBER(12)      NOT NULL,
O_CUSTKEY           NUMBER(12)      NOT NULL,
O_ORDERSTATUS       CHAR(1)         NOT NULL,
O_TOTALPRICE        NUMBER(12,2)    NOT NULL,
O_ORDERDATE         DATE        NOT NULL,
O_ORDERPRIORITY     CHAR(15)    NOT NULL,
O_CLERK             CHAR(15)    NOT NULL,
O_SHIPPRIORITY      NUMBER(12)      NOT NULL,
O_COMMENT           VARCHAR(79)     NOT NULL,
      CONSTRAINT ORDERS_PKEY PRIMARY KEY (O_ORDERKEY),
      CONSTRAINT ORDERS_FKEY1 FOREIGN KEY (O_CUSTKEY)
            REFERENCES CUSTOMER(C_CUSTKEY),
      CONSTRAINT ORDER_CHECK1 CHECK(O_TOTALPRICE >= 0) );

CREATE TABLE LINEITEM(
L_ORDERKEY          NUMBER(12)      NOT NULL,
L_PARTKEY           NUMBER(12)      NOT NULL,
L_SUPPKEY           NUMBER(12)      NOT NULL,
L_LINENUMBER        NUMBER(12)      NOT NULL,
L_QUANTITY          NUMBER(12,2)    NOT NULL,
L_EXTENDEDPRICE     NUMBER(12,2)    NOT NULL,
L_DISCOUNT          NUMBER(12,2)    NOT NULL,
L_TAX               NUMBER(12,2)    NOT NULL,
L_COMMENT           VARCHAR(44)     NOT NULL,
      CONSTRAINT LINEITEM_PKEY PRIMARY KEY (L_ORDERKEY, L_LINENUMBER),
      CONSTRAINT LINEITEM_FKEY1 FOREIGN KEY (L_ORDERKEY)
            REFERENCES ORDERS(O_ORDERKEY),
      CONSTRAINT LINEITEM_CHECK1 CHECK (L_QUANTITY >= 0),
      CONSTRAINT LINEITEM_CHECK2 CHECK (L_EXTENDEDPRICE >= 0),
      CONSTRAINT LINEITEM_CHECK3 CHECK (L_TAX >= 0),
      CONSTRAINT LINEITEM_CHECK4 CHECK (L_DISCOUNT BETWEEN 0.00 AND 1.00));
```

a) Consider the customer's order application database described in the previous question.

(i) Implement a **stored** PL/SQL function

customerName (clerkName)

that finds the names of all customers (c_name) who have made more than 1 orders and the orders were processed by a specific clerk (o_clerk). The function should return a string of customer names separated with a comma as shown below. Assume that a name of the clerk is passed to the function as a value of parameter clerkName. A sample output is as shown below:

Names of customers whose orders were processed by Clerk#000000996:
Customer#000036398, Customer#000028261, Customer#000039859

(5.0 marks)

(ii) Use the function customerName to list the names of all customers (c_name) whose orders were processed by a clerk 'Clerk#000000996' (o_clerk). You do not have to actually execute the function. You just need to write out the select statement. **(1.0 marks)**

b) Total number of orders made by a customer (C_TOTORDERS) is stored in the relational table CUSTOMER. The value is the sum of all orders made by a customer. Implement a row level trigger that automatically changes or update the total number of orders for each customer whenever a customer has made an order, remove an incorrect order, or changes to the orders made, that is, an order is added or inserted or updated in the relational table ORDERS. **(4.0 marks)**

END OF Question 3