SIM GLOBAL EDUCATION

UNIVERSITY OF WOLLONGONG AUSTRALIA

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 2 - (Total 10 marks)
Indexing

Time allocated: 40 minutes
Start time: 3:10 pm SGT
End time: 3:50 pm SGT
Submission time start: 3:45 pm SGT
Submission time end: 4:00 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));
```

Determine what index should be created to improve the performance of the queries listed below in the best possible way. **Consider each one of the queries as an individual case**. If you decide that an index should be created, then **list** the names of attributes that form an index key and **write** the **'create index'** statement to create the index. Remember that the order of attributes in an index key is important. **Explain** why a new index is needed **or** is not needed to improve the performance of the queries. That is, if a new index is created, explain why the index improve the performance of the query, and if no new index is created, explain why the performance of the query is not suffer (degraded). Assume that all relational tables are large enough to make full tables scan more time consuming than accessing the tables through an index.

i.
```
SELECT C_CUSTKEY, C_NAME, C_ADDRESS
FROM CUSTOMER
ORDER BY C_NAME;
```
(2.0 marks)

ii.
```
SELECT C_ADDRESS, COUNT(*)
FROM CUSTOMER
WHERE C_NAME = 'James Bond'
GROUP BY C_ADDRESS
HAVING COUNT(*) > 2
ORDER BY C_ADDRESS;
```
(2.0 marks)

iii.
```
SELECT DISTINCT O_TOTALPRICE, O_ORDERDATE
FROM ORDERS
ORDER BY O_ORDERDATE;
```
(2.0 marks)

iv.
```
SELECT L_PARTKEY, COUNT(*)
FROM LINEITEM
GROUP BY L_PARTKEY
HAVING COUNT(L_TAX) > 4;
```
(2.0 marks)

v.
```
SELECT O_CLERK, COUNT(O_CUSTKEY)
FROM ORDERS
WHERE TO_CHAR(O_ORDERDATE, 'YYYY') = '2021'
GROUP BY O_CLERK;
```
(2.0 marks)

END OF Question 2