

## School of Computing and Information Technology

### Student to complete:

Family name	WONG
Other names	ZUO WEI
UOW Student number	6461967

## CSCI235 Database Systems

### Final Examination Paper Session 2 2020 3 June 2020

Exam duration	3 hours and 10 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 / typed neatly.

**No asking for explanation of question is allowed during the examination. However, if you suspect that there is a typo or incorrect question, you can ask the invigilator to confirm.**

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

**Version 1.1**

## Question 2 - (Total 8 marks) Indexing

**Time allocated: 30 minutes**

**Start time: 10:45 am SGT**

**End time: 11:15 am SGT**

**Submission time start: 11:10 am SGT**

**Submission time end: 11:20 am SGT**

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

```
CREATE TABLE BOOK (  
  B_ISBN          NUMBER(15)          NOT NULL, /* Unique ISBN of book */  
  B_TITLE         VARCHAR2(15)        NOT NULL, /* Title of book */  
  B_PUBDATE       VARCHAR2(15)        NOT NULL, /* Publication date of book */  
  B_SUBJECT       VARCHAR2(30)        NOT NULL, /* Subject of book */  
  B_COST          NUMBER(6,2)         NOT NULL, /* Cost of book */  
  CONSTRAINT BOOK_PKEY PRIMARY KEY (B_ISBN) );
```

In addition to the primary key index, the administrator has created an additional index bookIdx(B\_PUBDATE, B\_SUBJECT, B\_COST) over the relational table BOOK.

- a) Find SELECT statements that will use the indexes in the ways specified in the question (i) to (v) below. The values used to create the query such that the queries can meet the specified criteria is up to you.

- (i) Execution of the first SELECT statement must traverse the index **vertically** and it **must not** access the relational table BOOK.

**(1.0 mark)**

**Select count(\*)  
from book  
where b\_pubdate = "any date";**

- (ii) Execution of the second SELECT statement must traverse the index **vertically** and later on **horizontally** and it **must not** access the relational table BOOK.

**(1.0 mark)**

**Select \*  
From book**

Where **b\_pubdate > "some date"**;

- (iii) Execution of the third SELECT statement must traverse the **leaf level** of the index **horizontally** and it **must not** access the relational table BOOK.

**(1.0 mark)**

**Select b\_pubdate**  
**From book**

- (iv) Execution of the fourth SELECT statement must traverse the index **vertically** and it **must** access the relational table BOOK.

**(1.0 mark)**

**Select \***  
**From book**  
**Where b\_pubdate > "some date"**  
**and b\_subject = "some name"**  
**and b\_cost = "some value";**

- (v) Execution of the fifth SELECT statement must traverse the index **vertically** and later on **horizontally** and it **must** access the relational table BOOK.

**(1.0 mark)**

**Select \***  
**From book**  
**Where b\_pubdate = "some date"**  
**And p\_cost = "some value"**

- b) Consider the following SELECT statement:

```
SELECT B_TITLE, COUNT(*)  
FROM BOOK  
WHERE B_SUBJECT = 'Database Design'  
AND B_COST = 70.0  
GROUP BY B_TITLE
```

```
HAVING COUNT(*) > 2  
ORDER BY B_TITLE;
```

- (i) **Find** the **best** index based on a single column (or single attribute) to speed up the processing of the query given above (Q2b). **Write** an SQL 'create index' statement to create the index. **Write** a brief explanation on how the index on a **single column** is used when the query is processed.
- (1.0 mark)**

**Create index bookSubjectIDX on book (b\_subject)**

**User is querying b\_subject. The database will use bookSubjectIDX when searching for "database design"**

- (ii) **Find** the **best** composite index based on two columns (or two attributes) to speed up the processing of the query given above (Q2b). **Write** an SQL 'create index' statement to create the index. **Write** a brief explanation on how the index on **two columns** is used when the query is processed.
- (1.0 mark)**

**Create index bookTitleSubjectIDX on book (b\_subject, b\_cost)**

**User is looking for a certain subject with a cost restriction. The Index will group the subjects and then create another subgroup with the cost accordingly**

- (iii) **Find** the **best** composite index based on three columns (or three attributes) to speed up the processing of the query given above (Q2b). **Write** an SQL 'create index' statement to create the index. **Write** a brief explanation on how the **composite index** is used when the query is processed.
- (1.0 mark)**

**Create index bookTitleSubjectIDX on book (b\_subject, b\_title, b\_cost)**

**The database will search for the subject, group the title and then organise them by cost, thus drastically improving the performance of the query.**

---

*End of specification*

**Answer:**

**Answer:**

**Answer:**