Section:

ICOM 6005 – Database Management Systems Exam II Due: December 3, 2001, 11:59 PM

Name: _____

Student Number: _____

Instructions:

- 1. Write your name on all pages of this exam.
- 2. Answer this exam, and send it via e-mail as a Word or PDF document.
- 3. The Web page contains extra slides with the topics discussed in class.
- 4. There are five problems and a bonus in this exam.
- 5. Read each question carefully, and show all the work you used to generate your answer.
- 6. To receive partial credit, you **MUST SHOW** all the work you used to generate your answer.

GOOD LUCK!

Section:

Score

1	/20
2	/20
3	/20
4	/20
5	/20
Bonus	/10
Total	/100

Section:

Problem 1. (20 points) RAIDS (Seed Slides on Web Page and Chapter 7 of text)

Explain what are the RAID Level 10, RAID Level 4 and RAID Level 5 organizations. Compare each one in terms of effective space utilization, and cost for reads and writes. Keep your answer up to two pages (use the space being provided in this exam).

Section:

Problem 1 (Continuation)

Section:

Problem 2. (20 points) Buffer Management (see Chapter 7 of textbook, chapter 3 Old one)

Explain each of the following terms:

a) Buffer pool

b) Pinning a page

Section:

Problem 2 (Continuation)

c) Dirty Page

d) Least Recently Use replacement Policy

Section:

Problem 3. (20 pts) Record formats (See Chapter 7 of the textbook, Chapter 3 old one)

There are two principal record formats used to represent variable-length records. These are:

- 1. **Delimited** In the record, each attribute is separated by a special character
- 2. Array of Field Offsets At the beginning of the record, there is an array that indicates the offset (number of bytes to be skipped) to reach the beginning of each attribute.

Explain the advantages and disadvantages of each one. Which one is more efficient in terms of the effort needed to find a given attribute within the record? Keep your answer brief.

Section:

Problem 3 (Continuation)

Name:

Section:

Problem 4 (20 points) Page Formats (see Chapter 7 of textbook, chapter 3 old one)

Consider the following two page formats used to represent fixed-length records:

- 1. **Packed** The page has a 4-byte integer at the end of the page indicating the number of records **currently** stored in the page. The rest of the space is used to store as many records as possible, and each record has the same size (in bytes).
- 2. Unpacked-Bitmap The page has a 4-byte integer at the end of the page indicating the number of records that **can be stored** in the page. In addition, there is an array of 1-byte characters, used to indicate if a given record within the page is valid or not. If entry *n* of the array is 1, then record *n* is valid; otherwise record *n* is not valid (it was either deleted or never being used).

Explain the advantages and disadvantages of each page organization in terms of their effective space utilization and the complexity involved in deleting a record from a page.

Section:

Problem 4 (Continuation)

Section:

Problem 5. (20 points) Join Processing (see chapter 12 of textbook)

a) Compare the tuple-at-a-time Nested Loops join with the page-at-a time Nested Loops join in terms of their efficiency and complexity of implementation.

Section:

Problem 5. (Continuation)

b) Suppose that we have two relations **Students** and **Courses**, with the following information:

Relation Name	Number of Tuples	Number of Pages
Students	20,000	7,500
Courses	2,300	890

What is the cost of computing the following joins?

- 1. Students ⊳⊲Courses
- **2.** Courses ⊳⊲ Students

Assume that a tuple-at-a-time nested loops join algorithm is used, and that each I/O has a cost of 10msec.

Section:

Bonus (10 points) Join Processing (chapter 12 of textbook)

Suppose that we have three relations **Students**, **Courses**, and **Teachers**, each with the following information:

Relation Name	Number of Tuples	Number of Pages
Students	20,000	7,500
Courses	2,300	890
Teachers	5,000	1,300

Suppose that we need to compute a three-way equi-join (join of three tables) between tables **Students**, **Courses**, and **Teachers**. If a tuple-at-a-time Nested Loops Join algorithm is used, and the I/O cost is 10msec, what is the most efficient order to execute the joins between the relations? Justify your answer based on the cost of each possible join ordering.

Section:

Bonus (Continuation)