UNIVERSITY OF CALIFORNIA
Department of EECS, Computer Science Division

CS186 Final Exam
May 16, 2000

Final Exam: Introduction to Database Systems

This exam has seven sections, each with one or more problems. Each problem may be made up of multiple questions. You should read through the exam quickly and plan your time-management accordingly. Before beginning to answer a question, be sure to read it carefully and to *answer all parts of every question!*

You **must** write your answers on **these stapled pages**. You also must **write your name** at the top of **every page except this one**, and you must turn in all the pages of the exam. You may remove this page from the stapled exam, to serve as a reference, but **do not remove any other pages from the stapled exam**! Two pages of extra answer space have been provided at the back in case you run out of space while answering. If you run out of space, be sure to make a "forward reference" to the page number where your answer continues.

REFERENCE DATABASE . This is the Reference Database referred to in some of the questions.

There are six tables describing a company, describing employees, departments, buildings, which department(s) an employee works in (and a percentage of the time for each), department managers (possibly more than one per department), and in which building an employee works (an employee may have more than one office). The primary key of each table is the attribute(s) in capitals. Other attributes are not necessarily unique.

EMP – 100,000 tuples, 1,000 pages

EID	EName	Salary	Start_Date	End_Date
001	Jane	$124,000	3/1/93	null
002	Jim	$32,000	2/29/96	null
003	John	$99,000	12/12/98	null
004	Joe	$55,000	2/2/92	null
005	Jenny	$51,000	5/5/95	null

EID values range from 1 to 100,000

IN_DEPT – 110,000 tuples, 550 pages

EID	DID	Percent_Time
001	101	100
002	102	100
003	101	60
003	102	40
004	103	100
005	103	100

BUILDING – 2,000 tuples, 10 pages

BID	BName	Address
201	ATC	1600 Ampitheatre
202	CCC	500 Crittenden
203	MFB	123 Shoreline

BID values range from 1 to 2,000

IN_BUILDING – 110.000 tuples, 550 pages

EID	BID
001	201
002	201
003	202
003	203
004	202
005	203

DEPT – 1,000 tuples, 5 pages

DID	DName	Annual_Budget
101	Research	$1,001,000
102	Development	$500,000
103	Sales	$2,000,000

DID values range from 1 to 1000

MANAGES_DEPT – 800 tuples, 4 pages

EID	DID
003	101
003	102
001	103

Name_____ **Login**_____

I. **SQL** – All queries are based on the sample schema shown on the first page. Assume that the tables have many more rows than are shown there. 15 Points.

1. Which of the following queries finds the names of buildings where more than 50 employees work? **(Circle as many as are correct.) (5 points)**

   a.
   ```
   SELECT Bname
   FROM IN_BUILDING
   GROUP_BY BID
   WHERE Count(*) > 50
   ```

   b.
   ```
   SELECT Bname
   FROM BUILDING
   WHERE BID IN (SELECT BID FROM In_Building
                 GROUP BY BID HAVING Count(*) > 50)
   ```

   c.
   ```
   SELECT Bname
   FROM Building B, In_Building I
   WHERE B.BID = I.BID
   GROUP BY B.BID
   HAVING Count(*) > 50
   ```

   d.
   ```
   SELECT Bname
   FROM Building B
   WHERE 50 < (SELECT Count(*) FROM In_Building I
               WHERE I.BID = B.BID)
   ```

   e. None of the above

2. Which of the following queries finds the name of Departments where no employees work? **(Circle as many as are correct.) (5 points)**

   a.
   ```
   SELECT Dname
   FROM Dept
   WHERE DID IN (SELECT I.DID FROM In_Dept I
                 GROUP BY I.DID HAVING COUNT(*) = 0)
   ```

   b.
   ```
   SELECT Dname
   FROM Dept D, In_Dept I, Emp E
   WHERE I.EID = E.EID and D.DID = I.DID and Count(E.EID) = 0
   ```

   c.
   ```
   SELECT Dname
   FROM Dept
   WHERE DID NOT IN (SELECT DISTINCT DID FROM In_Dept I)
   ```

   d.
   ```
   SELECT Dname
   FROM Dept D
   Where Not Exists (SELECT * FROM In_Dept I, EMP
                     WHERE I.EID = EMP.EID and I.DID = D.DID)
   ```

   e. None of the above

Points for this page: _____

Which of the following queries finds the name of the Department(s) where the highest paid employee works? **(Circle as many as are correct.) (5 points)**

f.
```
SELECT D.Dname
FROM Dept D
WHERE D.DID IN (SELECT T.DID, MAX(Salary) FROM Dept T, In_Dept I, Emp E
                    WHERE T.DID = I.DID and E.EID = I.EID)
```

g.
```
SELECT Dname
FROM Dept D, In_Dept I, Emp E
WHERE D.DID = I.DID and E.EID = I.EID and E.Salary = MAX(Salary)
```

h.
```
SELECT DName
FROM Dept D, In_Dept I, Emp E
WHERE D.DID = I.DID and E.EID = I.EID and E.Salary >= ALL
    (SELECT Salary FROM EMP)
```

i.
```
SELECT DName
FROM Dept
WHERE DID IN
            (SELECT I.DID FROM In_Dept I, Emp E
             WHERE I.EID = E.EID AND
                    E.Salary = (SELECT MAX(Salary) FROM Emp))
```

j. None of the above

Implementation of Relational Operators – 18 points

Consider the schema on the first page, and the number of tuples and pages for each relation shown there. Let "×" be the join operator, and "A×B" means join with A as the outer relation and B as the inner.

As we did in class, when computing the cost for join algorithms, you may ignore output cost (since this is the same for all algorithms).

Note: you have 9 pages of main memory to work with in these problems.

1. Consider the operation: $\sigma_{(EID < 5000)}$**EMP** (2 points)

   a) What is the I/O cost of this operation? ***1000***

   b) What is the reduction factor? ***0.05 = 1/20***

2. Consider the join: **In_Dept × Dept** (4 points)

   a) What is the I/O cost of this using Blocked Nested Loops? ____

   945 = 550 + Ceiling(550/7)*5

   b) What is the I/O cost of this using Index Nested Loops, with a Hash index on Dept.DID?

   242,550 = 550 + 110,000 * 2.2

3. Consider the join: **Dept × In_Dept** (4 points)

   a) What is the I/O cost of this using Blocked Nested Loops?

   555 = 5 + Ceiling(5/7)*550

   b) What is the I/O cost of this using Index Nested Loops, with a Hash index on In_Dept.DID? ***I accepted both:***

   2205 = 5 + 1000*2.2

   242,005 = 5 + 1000 * 2.2 * 110 (the approx #matches is 110)

4. Consider the join**: EMP × In_Building (**8 points)

   a) What is the I/O cost of this using Blocked Nested Loops?

   79650 = 1000 + Ceiling(1000/7)*550

   b) What is the I/O cost to sort EMP?

   8000 = 2*1000(Ceiling(Log8(1000/9)) + 1)

   c) What is the I/O cost to sort In_Building?

   3300 = 2*550(Ceiling(Log8(550/9)) + 1)

   d) What is the total I/O cost to do this using Sort/Merge join?

   12850 = 8000 + 3300 + 1000 + 550

Points for this page: _____

II. Query Optimization – 13 points

Consider the schema shown on the first page and especially the number of tuples and pages for each relation.

Consider the following query:

```
Select  Bname
From EMP E, Building B, In_Building I
Where E.EID < 500 and E.EID = I.EID and B.BID = I.BID
```

1. Write this query in relational algebra. (3 points)

$\Pi_{\text{Bname}}\ \sigma_{(\text{EID} < 500)}\ (\text{Emp} \times \text{In\_Building} \times \text{Building})$

2. If the database has an unclustered B-Tree index on EMP.EID, what is the best plan you can find to execute this query? Do your work on the additional pages at the back of the exam, and show the query plan here, including the costs for each step and the total cost. (10 points)

This was graded such that I looked for a plan tree, computation of costs at each step, and some consideration of reduction factors.

Points for this page: _____

Name_____ Login_____

III. Normalization – 15 points

Consider the follow attributes and functional dependencies:

A B C D E F H
A→ D
AE → H
DF → BC
E → C
H → E

1. List all keys (**not** superkeys): (3 points)

AEF
AFH

2. Which of the following dependencies are implied by those above: (4 points)
   a. (IS) (IS NOT) A → AD
   b. (IS) (IS NOT) A → DH
   c. (IS) (IS NOT) AED → C
   d. (IS) (IS NOT) DH → C
   e. (IS) (IS NOT) ADF → E

3. Consider the decomposition into 4 relations: (AEH) (ABCF) (AD) (CE). Is this decomposition in (circle all that apply): (4 points)
   a. BCNF
   b. 3NF
   c. 1NF
   d. None of the above

4. Consider the decomposition into 3 relations: (AD) (EC) (ABEFH). Is this decomposition in (circle all that apply): (4 points)
   a. BCNF
   b. 3NF
   c. 1NF
   d. None of the above

Points for this page: _____

IV. Concurrency Control and Crash Recovery: LOCKING – 15 Points

Locking is the most popular concurrency control technique implemented by commercial database management systems.

1. Consider a database that is read-only (i.e., no transactions change any data in the database, data may be loaded into the database when the database is off-line). Suppose serializability needs to be supported. Please circle all correct statements: (5 points)

   a. No locking is necessary.
   b. Only read locks are necessary and they need to be held until end of transaction.
   c. Only read locks are necessary but they can be released as soon as the read is complete.
   d. Both read and write locks are necessary and locking must be done in two phases.
   e. None of the above.

Consider the following database schema:

STUDENT(name, sid, gpa, level, dept)

Suppose the following two transactions are executed concurrently:

T1:
```
begin tran
update STUDENT set gpa = 4.0 where dept = 'CS'
commit tran
```

T2:
```
begin tran
insert into STUDENT values ('Mihut', 101, 3.9, 4, 'CS')
insert into STUDENT values ('Sirish', 102, 3.9, 3, 'CS')
commit tran
```

2. Assume Mihut and Sirish were not in the STUDENT table before the start of T1 or T2. Suppose read locks are released immediately after the read is done and write locks are held until end of transaction. Can it ever happen that after both T1 and T2 have committed, Mihut and Sirish have different gpa values? Please state your reasoning in support of your conclusion. If your answer depends on locking granularity, access methods or indexing, please analyze the possibilities. (10 points)

This was an example of the Phantom Problem. If you only have row-level locking, Mihut & Sirish may end up with different GPAs. If you have table-level locking, they will end up with the same GPA.

Concurrency Control and Crash Recovery: WRITE-AHEAD LOGGING – 12 points

Write-ahead logging is the most popular recovery technique.

1. Checkpoint is a technique that can reduce recovery time after a crash. Please circle the correct statements: (4 points)

   a. After a soft crash (which does not affect data on hard drives), the log only needs to be scanned back until the last checkpoint is found. The log beyond the last checkpoint will not be read during the recovery process.
   b. Once a checkpoint is done, the log can be truncated.
   c. Checkpoint is automatically performed after every transaction commit.
   d. Checkpoints should be done after every update to the database.
   e. None of the above.

2. This question deals with when updated data pages (dirty pages) must be written to disk. Please circle the correct statements: (4 points)

   a. Updated pages must be written to disk immediately after the update.
   b. Dirty pages must be written to disk at transaction commit time but before the transaction log is written to disk.
   c. Dirty pages must be written to disk at transaction commit time but after the transaction log is written to disk.
   d. A dirty page must be written to disk when it is replaced from the buffer pool.
   e. None of the above.

3. Since a database log can grow without limits, the log should be truncated at some point. Where can the log be truncated? (4 points)

Mihut will provide the key to this question.

V. Concurrency Control and Crash Recovery: RECOVERY – 12 points

1. If the buffer pool is large enough that uncommited data are never forced to disk, is UNDO still necessary? How about REDO? (4 points)

   UNDO

   a) YES
   b) NO

   REDO

   c) YES
   d) NO

2. If updates are always forced to disk when a transaction commits, is UNDO still necessary? How about REDO? (4 points)

   UNDO

   a) YES
   b) NO

   REDO

   c) YES
   d) NO

3. With checkpoint, after a softcrash, where in the log should REDO start? Where should UNDO start? (4 points)

   a) REDO:

   At the smallest recLSN in the dirty page table.

   b) UNDO:

   At the last LSN for each transaction to be undone.

Points for this page: _____

Extra Credit – 8 points

- **Broadbase: Data Marts & OLAP (2 points)** - The presenter from BroadBase described how their database uses "Cubes", pre-computed indexes of aggregate information. In 15 words or less, what aspect of the workload allows them to use "Cubes"?

A read-only workload allows them to use cubes, which otherwise would be very expensive.

- **Evite: Managing data at a Web Site (5 points)** - The presenter from E-Vite expressed opinions on the following topics w.r.t. their DBMS. In 10 words or less, what were her opinions of:
  i. Primary Keys
  They don't use them because they slow the database.
  ii. Integrity Constraints
  They don't use them because they slow the database.
  iii. Blobs
  They keep them separate because they slow the database.
  iv. Indexes
  Very important for making the database fast.

  E-vite uses the log for an interesting purpose not discussed in the book. In 15 words or less, what unusual thing do they do with the log?

They copy the log from one machine to another to keep a backup DBMS always running, ready, and up-to-date.

- **MineSet: Data Mining (1 point)** - Name a data mining algorithm mentioned in the guest lecture on Data Mining:

Decision Trees, Clustering, Neural Nets