

Name _____

Class Account:

UNIVERSITY OF CALIFORNIA
Department of EECS, Computer Science Division

CS186
Fall 2006

Hellerstein/Olston
Midterm Exam

Midterm Exam: Introduction to Database Systems

This exam has four problems and one extra credit question, worth different amounts of points each. Each problem is made up of multiple questions. You should read through the exam quickly and plan your time-management accordingly. Before beginning to answer a problem, be sure to read it carefully and to *answer all parts of every problem!*

You **must** write your answers on the 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. ***Do not tear pages off of your exam!***

Good luck!

1. Relational Query Languages [20 points]

Recently, Bob the Builder coded up a Relational Query Language Translator (RQLT) and passed it along to his lesser-known colleague, Ted the Software Test Engineer, for testing. Fortunately for Ted, the RQLT is simple and only operates on a single schema:

Tool(tid, brand, cost)

Jobsite(location, compensation, task)

Toolbox(tbid, location) – location is a foreign key to Jobsite

Holds(tbid, tid) – tbid is a foreign key to Toolbox, tid is a foreign key to Tool.

- a. (5 points) Ted sets the RQLT to translate the following SQL query to Relational Algebra:

```
SELECT T.tid
FROM Tool T, Holds H, Toolbox B, Jobsite J
WHERE T.tid = H.tid AND H.tbid = B.tbid
AND B.location = J.location AND J.task = 'Plumbing'
```

Which of the following would be an equivalent Relational Algebra query?

1. $\pi_{tid}(Tool \bowtie Holds \bowtie Toolbox \bowtie \sigma_{task = 'Plumbing'}(Jobsite))$
2. $\pi_{tid}(\sigma_{task = 'Plumbing'}(Tool \bowtie (Holds \bowtie Toolbox) \bowtie Jobsite))$
3. $\sigma_{task = 'Plumbing'}(\pi_{tid}(Tool) \bowtie Holds \bowtie Toolbox \bowtie Jobsite)$
4. 1 and 2
5. 1, 2, and 3
6. None of the above

YOUR ANSWER HERE (1a): _____

Name _____

- b. (10 points) Ted switches the RQLT to translate Relational Algebra into Relational Calculus and passes in:

$$\pi_{tbid}(Toolbox) - \pi_{tbid} ((\pi_{tbid}(Toolbox) \times \pi_{brand}(Tool)) - \pi_{tbid,brand}(Holds \bowtie Tool))$$

Fill in the blanks to complete the equivalent Relational Calculus Query. *Note:* the correct answer may not require all blanks to be filled in.

$$\{ B1 \mid B \in Toolbox \text{ _____ } B.tbid = B1.tbid \text{ _____} \\ \text{ _____ } T \in Tool \\ (\text{ _____ } H \in Holds (T.tid = H.tid \text{ _____ } H.tbid = B.tbid)) \}$$

- c. (5 points) Ted thinks that he has found a bug in the RQLT. It translates the following Relational Calculus query:

$$\{ H1 \mid H \in Holds \wedge H1.tbid = H.tbid \wedge \\ \forall T \in Tool ((H.tid = T.tid) \\ \Rightarrow (T.brand = 'Craftsman' \vee T.brand = 'Channellock')) \}$$

into the following incorrect SQL query (marked with line numbers for the purpose of this question):

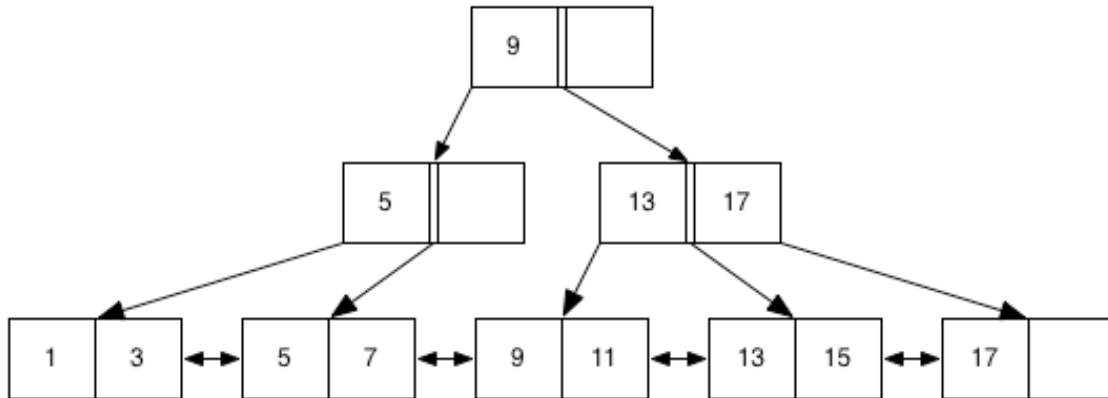
```
1. SELECT H.tbid
2.   FROM Tool T, Holds H
3.   WHERE T.tid = H.tid AND T.brand = 'Craftsman'
4. INTERSECT
5. SELECT H1.tbid
6.   FROM Tool T1, Holds H1
7.   WHERE T1.tid = H1.tid AND T1.brand = 'Channellock';
```

Fix this SQL query to match the Relational Calculus query by changing exactly ONE line:

Answer: Change line _____ into:

2. B+ Trees [23 points]

Consider the following B+ tree index of order 1:



- a. (3 points) Circle all nodes (not index entries, but entire nodes) in the above figure that must be fetched to satisfy the query "Get all records with search key greater than or equal to 7 and less than 15".
- b. (15 points) Assume we modify the B+ tree by adding the following keys **in the following order:** **20, 27, 18, 30, 19**

In the answer-boxes below, each row refers to a key being inserted in order, and each column asks if the insertion of that key results in a split of particular nodes. Assume that when splitting up an odd number of entries, the left node gets one more than the right. Place a check mark (✓) in each box whose answer is "Yes". Blank boxes will be interpreted as "No". You may want to use the back of the previous page for scratch space.

Key	Leaf Node Split?	Non-Leaf Split?	Root Split?
20			
27			
18			
30			
19			

- c. (5 points) Suppose we were to insert all integers in the range 26 to 4112 inclusive (i.e. 26, 27, 28 ... , 4111, 4112) into the tree in part (a), one at a time. *At most* how many levels would the resulting B+-tree have? (Hint: You should not need to draw a B+-tree to figure this out!)

YOUR ANSWER HERE (2c): _____

3. Query Optimization [32 points]

Consider the following schema:

```
Guitars (gid, brand, price)
Players (pid, name, age)
LastPlayed (gid, pid, date)
```

And the following query:

```
SELECT P.name
FROM Guitars G, Played P, LastPlayed L
WHERE G.gid = L.gid AND P.pid = L.pid
      AND P.age < 25 AND G.brand = 'Gibson'
      AND G.price > 3000;
```

In the schema, Guitars.gid is the primary key of Guitars, Players.pid is the primary key of Players, and LastPlayed.gid and LastPlayed.pid are foreign keys referencing the primary keys of Guitars and Players (respectively). Assume that the data is evenly distributed, and that the following properties hold:

```
Players.age ranges from 10 to 85
Guitars.brand has 10 distinct values
Guitars.price ranges from 1,000 to 5,000
Guitars.gid has 1,000 distinct values
Players.pid has 1,000 distinct values
```

- a. (5 points) Compute the selectivity for each individual term in the WHERE clause. You must fill in each blank below!

G.gid = L.gid : _____

P.pid = L.pid : _____

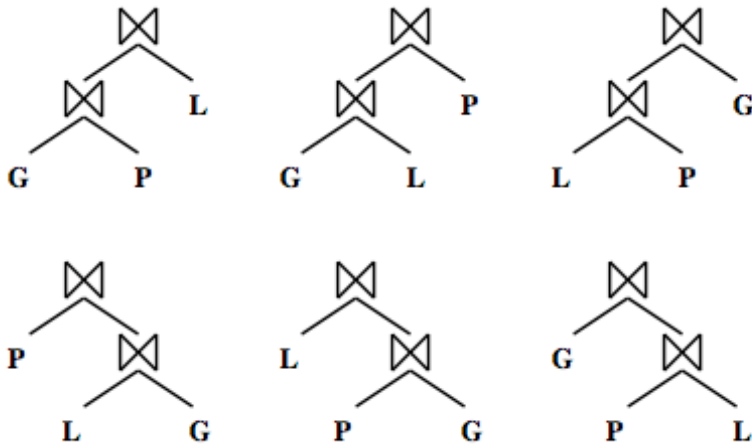
P.age < 25 : _____

G.brand = 'Gibson' : _____

G.price > 3000: _____

Name _____

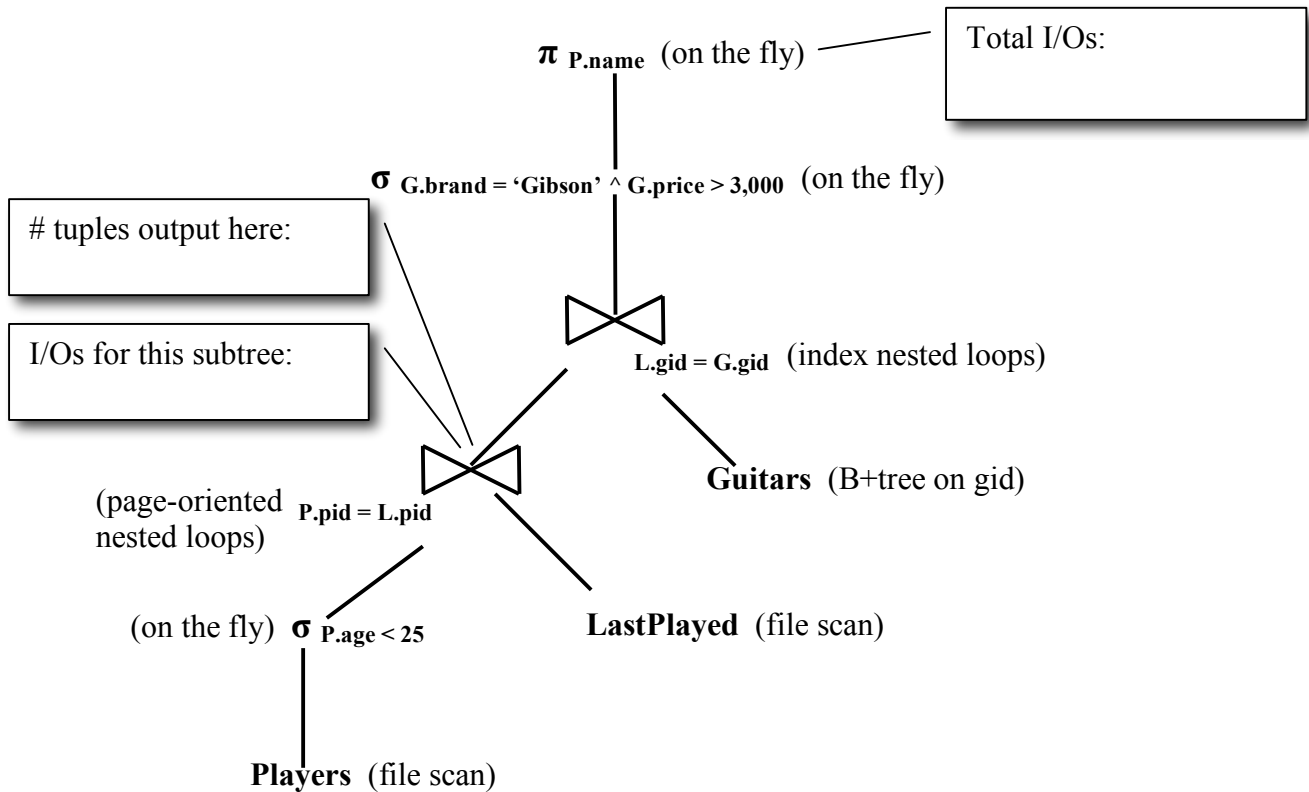
b. (5 points)
 According to the System R query optimizer that we studied, circle all the following join orders that would be considered:



c. (12 points) Now consider the following. There is a B+ tree index on Guitars.gid – it is unclustered and it uses Alternative 2 to represent data entries. Assume that, on average, it takes 3 I/Os to retrieve a given data entry in a leaf of the index, and that the following properties hold:

Guitars → 40 bytes/tuple, 100 tuples/page, 10 pages
 Players → 80 bytes/tuple, 50 tuples/page, 20 pages
 Lastplayed → 20 bytes/tuple, 200 tuples/page, 100 pages

Assuming *no buffering occurs*, fill in the three blanks in the diagram below.



Name _____

- d. (10 points) Consider the plan from part (c) again. For each of the following changes to it, mark the line “Y” (yes) or “N” (no) depending on whether it could have led to a further reduction in the number of I/Os. Consider each change individually! Assume that any cost associated with setting up the described change is *not* included in the execution cost, and that buffering is now being used.

_____ Pushing down the selection on G.brand and G.price below the join

_____ Creating a temporary file to store the results of the selection on P.age

_____ Having the index on Guitars.gid be clustered

_____ Projecting out P.age before the join

_____ Changing the first join to block-oriented nested loops join.

4. Query Execution [25 points]

You have been hired to advise a major hot-dog vending franchise called “Dunce Dog” to tune their database server installation. They run a commercial DBMS called Tentacle version 9y. One of the startup parameters in Tentacle is called `query_space`; it tells Tentacle how many disk-blocks worth of RAM it should allocate to use for sorting and hash joins. Note that memory used for `query_space` is separate from the Tentacle buffer pool.

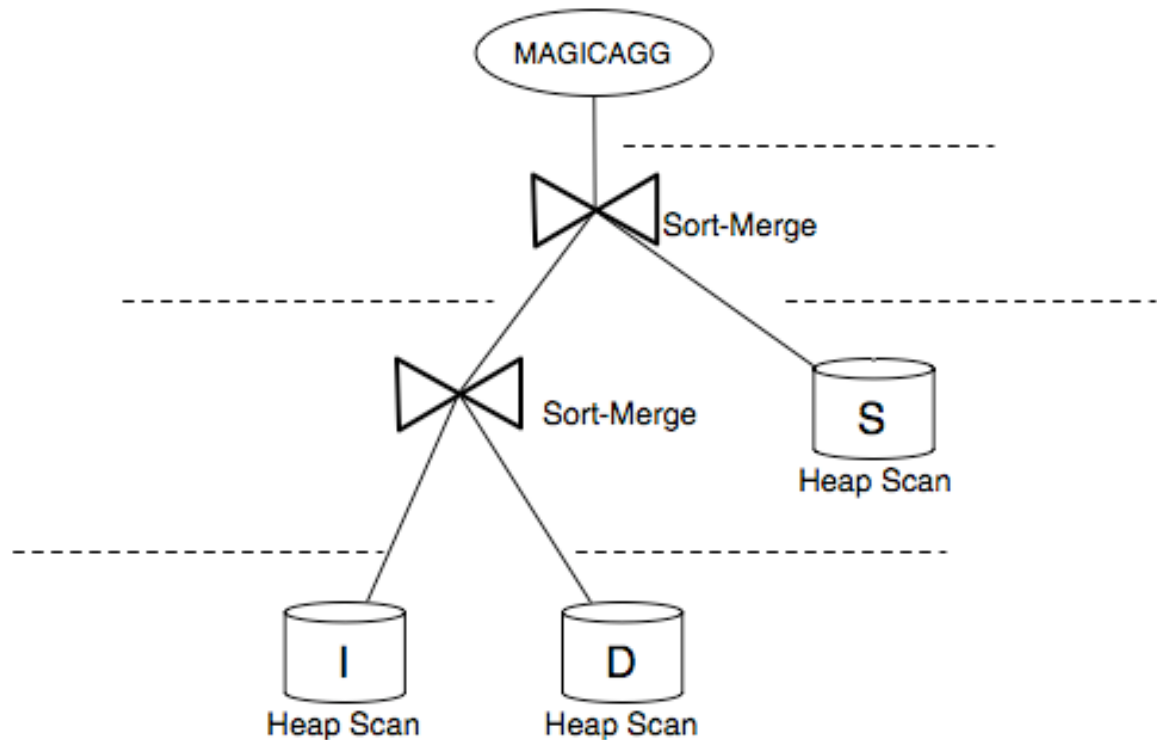
Dunce Dog is having problems running one of their monthly reporting queries. It uses the following tables:

Store(sid, location, owner)
 ItemsForSale(iid, name, description, cost, price)
 DetailedSales(receiptno, iid, sid)

The query is:

```
SELECT S.sid, S.location, SUM(I.price - I.cost) AS PROFIT
FROM Stores S, ItemsForSale I, DetailedSales D
WHERE S.sid = D.sid and D.iid = I.iid
GROUP BY S.sid, S.location
```

You determine that the query plan that Tentacle chooses is:



Name _____

- a) (5 points) Tentacle performs aggressive projection – that is, it discards any attributes that it does not need as soon as possible. For each edge in the query plan, write down the smallest list of attributes that needs to be *retained* (i.e. not discarded) on the corresponding dotted line.
- b) (5 points) Assume that the result of the scan and projection of ItemsForSale fits in 1000 pages, and the result of the scan and projection of DetailedSales fits in 10,000 pages. Approximately how many blocks of `query_space` should Tentacle need *at minimum* to perform the sort-merge join of ItemsForSale and DetailedSales in two passes? Feel free to round up or down by as many as 2 blocks in any equations that you use, but if you do so, show your equations!
- c) (5 points) Assume that the result of the first join fits in 2,500 pages, and the result of the scan of Stores fits in 400 pages. Approximating and showing work as in part (b), estimate the number of blocks of `query_space` that Tentacle should need to perform the second join in the plan in two passes.

Name _____

d) (5 points) How much memory should Tentacle's MAGICAGG operator need to perform GROUP BY and aggregation without I/O? Again, feel free to round up or down by as many as 2 blocks.

e) (5 points) Using the variables b , c and d to represent your answers to parts (b), (c) and (d) respectively, which of the following represents the minimum value of `query_space` that your client should use:

- a. $SUM(b, c, d)$
- b. $MAX(b, c, d)$
- c. $MIN(\sqrt{b}, \sqrt{c}, d)$
- d. $\log_{\text{query_space}} MAX(b, c, d)$

YOUR ANSWER HERE (4e): _____

Name _____

EXTRA CREDIT: The Disk Whisperer (10 points)

It's exhausting being a disk drive, constantly seeking, scanning, transferring... Since you are a sensitive Berkeley person, you are able to talk to your disk drive and it confides in you about how tough life is.

Recently, your disk drive told you that it's really tired of doing sort-merge joins. You offered to let it do hash joins all the time, even when the optimizer used to choose sort-merge join. Your disk responded, "Who cares – same stuff, different order."

| Explain why your disk drive said that.

Name _____

|

Name _____