Name: ____

CSE 444, Winter 2011, Final Examination

17 March 2011

Rules:

- Open books and open notes.
- No laptops or other mobile devices.
- Please write clearly and explain your reasoning
- You have 1 hour 50 minutes; budget time carefully
- An extra page is provided in case you run out of space, but make sure to put a forward reference.

Question	Max	Grade
1	27	
2	20	
3	34	
4	14	
5	5	
Total	100	
Extra credit	5	

Initials: ____________________________

1. (**27** points) **SQL and Relational Algebra**

   Consider the following schema:

```
Vehicle ( VIN, model, year )
Driver ( license, name, age )
Insured ( license, VIN, premium )
```

   The key fields are underlined: `VIN` is the key for `Vehicle`, `license` is the key for `Driver`, and `VIN` and `license` together form the key for `Insured`. Also, `Insured.license` is a foreign key that references `Driver.license`, and `Insured.VIN` is a foreign key that references `Vehicle.VIN`.

   (a) (**22** points) Write the following queries in Relational Algebra and SQL. For the RA part, you can give either the relational algebra expression, or the tree representation.

      i. (**6** points) Find the VINs of vehicles that are insured for a driver between 20 and 30 years of age.

         Relational Algebra:

         SQL:

ii. (**8** points) Find the VINs of vehicles that are insured for some driver under 25 years of age and another driver who is over 50 years of age.

Relational Algebra:

SQL:

Initials: ____________________________

iii. (**8** points) Some vehicles are operated by more than one driver, and a different premium may be charged for each driver for the same vehicle. Find pairs of license numbers such that the driver with the first license number gets charged a higher premium for the same vehicle compared to the driver with the second license number.

Relational Algebra:

SQL:

Initials: ___________________________

(b) (**5** points) For each one of the Relational Algebra (RA) expressions that follow, state what they compute. Simply select one of the statements below that correctly describes the output produced by each expression. **Different RA expressions may map to the same statement**, but one RA expression cannot correspond to multiple statements. For your convenience, the tree representations of the RA expressions are given on the next page.

Statements:

i. The models of vehicles insured by a driver under 25 years old for a premium of less than $200, and another driver who is over 50 years old, with a premium of less than $150.

ii. The models of vehicles insured by drivers under 25 years old for a premium of less than $200.

iii. The models of vehicles insured by a driver under 25 years old for a premium of less than $200, or a driver who is over 50 years old, with a premium of less than $150.

iv. The RA expression returns no tuples.

v. None of the statements i-iv correctly describes the output of the RA expression.

RA1: $\pi_{model}(\pi_{VIN}((\sigma_{age<25} Driver) \bowtie (\sigma_{premium<200} Insured)) \bowtie Vehicle)$

is described by statement: ___________

RA2: $\pi_{model}(\pi_{VIN}((\sigma_{age<25} Driver) \bowtie (\sigma_{premium<200} Insured) \bowtie Vehicle))$

is described by statement: ___________

RA3: $\pi_{model}(\sigma_{age<25 \wedge premium<200}((\pi_{VIN}(Driver \bowtie Insured)) \bowtie Vehicle))$

is described by statement: ___________

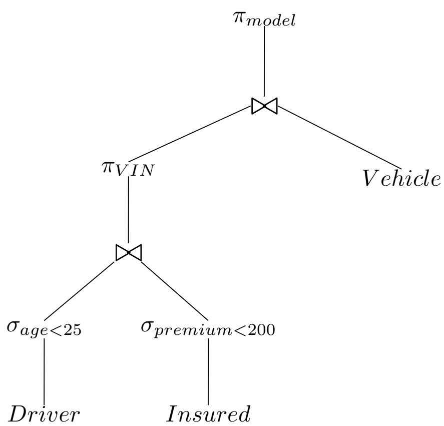
RA4: $(\pi_{model}((\sigma_{age<25} Driver) \bowtie (\sigma_{premium<200} Insured) \bowtie Vehicle)) \bigcup (\pi_{model}((\sigma_{age>50} Driver) \bowtie (\sigma_{premium<150} Insured) \bowtie Vehicle))$

is described by statement: ___________

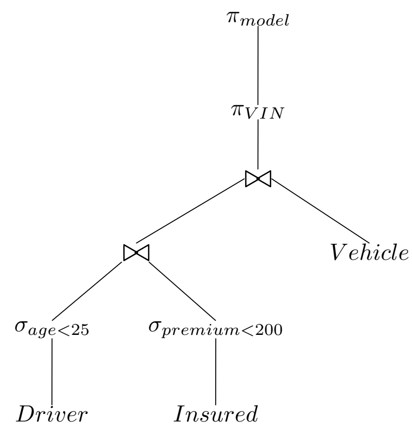
RA5: $\pi_{model}((\pi_{VIN,model}((\sigma_{age<25} Driver) \bowtie (\sigma_{premium<200} Insured) \bowtie Vehicle)) \bigcup (\pi_{VIN,model}((\sigma_{age>50} Driver) \bowtie (\sigma_{premium<150} Insured) \bowtie Vehicle)))$

is described by statement: ___________

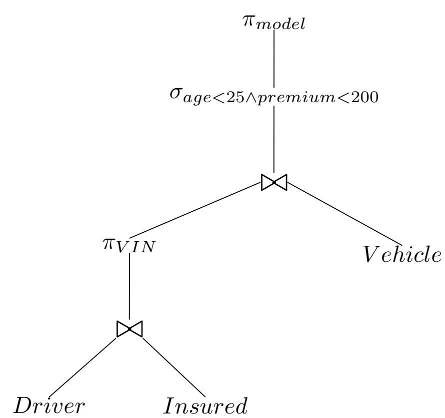
Initials: ____________________________



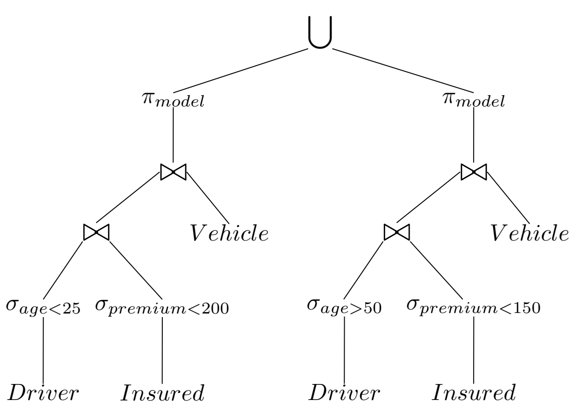
(a) RA1



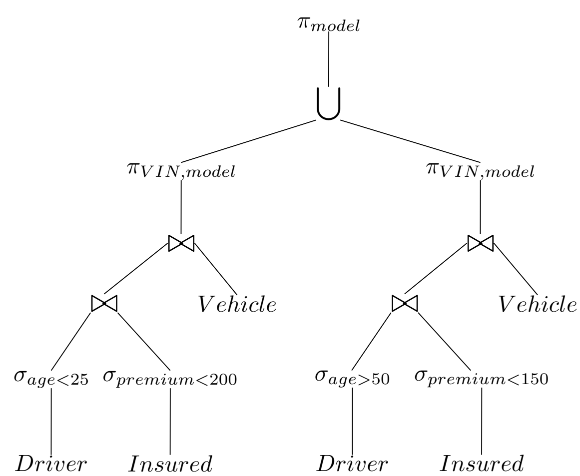
(b) RA2



(c) RA3



(d) RA4



(e) RA5

Initials: ____________________________

2. (**20** points) **Distributed Transactions**

Assume a distributed transaction T that was submitted at site 0 (coordinator), and spawned subtransactions at sites 1, 2, and 3 (subordinates). We describe the sequences of messages that can take place during the 2 Phase Commit Protocol as follows: Let `(i, j, M)` denote that site `i` sends message `M` to site `j`. The value of `M` can be one of the following:

```
P (prepare)
C (commit)
A (abort)
Y (yes vote, ready to commit)
N (no vote, do not commit).
```

For example, the message `(0, 1, P)` denotes a prepare message send from the coordinator to the first subordinate.

We ignore ack messages.

(a) (**2** points) Describe the sequence of messages send for a successful commit during the 2 phases of the 2PC protocol. At what point exactly is the transaction considered committed?

(b) (**2** points) Describe the sequence of 2PC messages in case site 3 does not want to commit the transaction. Underline any messages that can be omitted as an optimization.

Initials: ____________________________

(c) (**4** points) Suppose that the coordinator has sent all the prepare messages but has not yet received a vote from site 1. Would it be ok for the coordinator to abort the transaction at this point, and send abort messages to the subordinates? Why, or why not?

(d) (**4** points) Suppose that the coordinator has sent all the prepare messages, received a No vote from site 1, but has not yet received the votes of sites 2 and 3. Should the coordinator wait for the two missing votes, or should it proceed to abort the transaction and why?

(e) (**4** points) Suppose that site 1 has received a prepare message and voted Yes, but has not received any commit or abort messages. Site 1 contacts site 2 and discovers that site 2 has received a commit message. Is it ok for site 1 to commit the transaction? Why, or why not?

(f) (**4** points) Suppose that site 1 has received a prepare message and voted Yes, but has not received any commit or abort messages. Site 1 contacts all other subordinates and discovers that they have all voted Yes to the coordinators Prepare message. Is it ok for site 1 to commit the transaction? Why, or why not?

Initials: ______________________________

3. (**34** points) **Query Optimization**

Consider the following schema:

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

Here, `LastPlayed.gid` is a foreign key that references `Guitars.gid`, and `LastPlayed.pid` is a foreign key that references `Players.pid`. Consider 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;
```

Further assume that the data is evenly distributed, and that the following statistics hold:

`Guitars.gid` has 1,000 distinct values
`Guitars.brand` has 15 distinct values
`Guitars.price` ranges from 1,000 to 4,999
`Players.pid` has 15,000 distinct values
`Players.age` ranges from 11 to 85

(a) (**6** points) Compute the selectivity for each individual term in the where clause, then provide a final selectivity estimate for what fraction of the total tuples will appear in the output. Calculate each of the 6 values below:

`G.gid = L.gid` : ______________________________

`P.pid = L.pid` : ______________________________

`P.age <= 25` : ______________________________

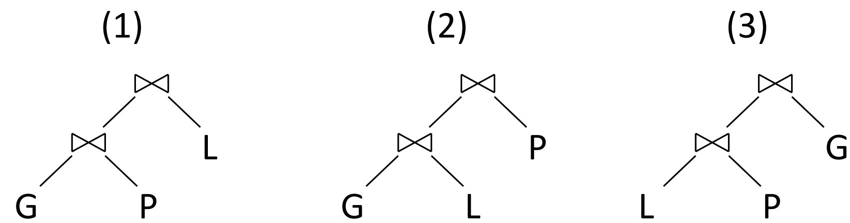
`G.brand = 'Gibson'` : ______________________________

`G.price >= 3000` : ______________________________

TOTAL : ______________________________

Initials: ____________________________

(b) (**2** points) In the following picture you see three possible linear plans (they are all left-deep). Which of the shown query plans would not be considered by any reasonable query optimizer and why? Circle the plan(s) that would not be considered and explain your reasoning below.

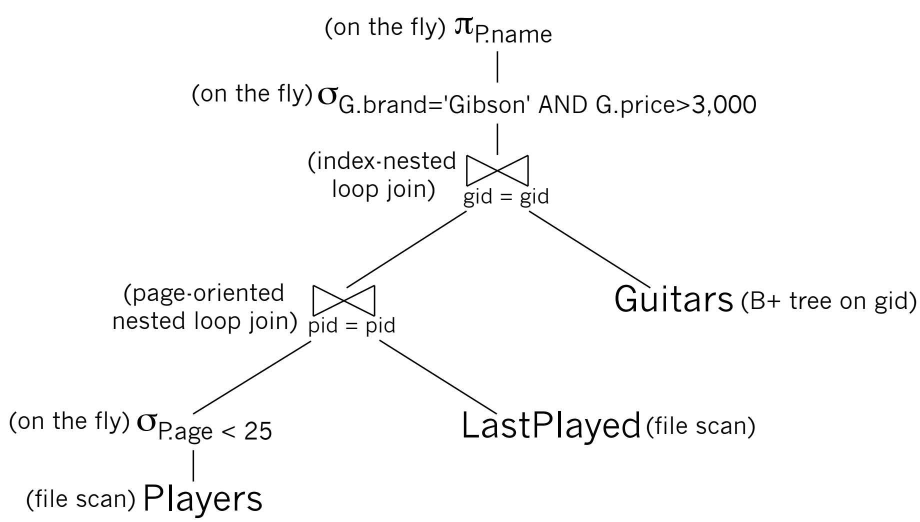


Initials: ______________________________

(c) (**16** points) Next assume there is an unclustered B+ tree index on `Guitars.gid` which is kept in main memory, and consider the following statistics for each of the 3 relations:

`Guitars`: 40 bytes/tuple, 100 tuples/page, size: 10 pages
`Players`: 80 bytes/tuple, 50 tuples/page, size: 300 pages
`LastPlayed`: 25 bytes/tuple, 150 tuples/page, size: 90,000 pages

Consider the query plan below. Calculate the number of I/Os, step-by-step. Make your calculations explicit so you can get partial credit. Remember that for a page-oriented nested loop, only one single page of the build relation is kept in main memory while iterating over the probe relation. Assume the main memory has clearly space for more than 2 pages.



Initials: ____________________________

(d) (**10** points) Consider the plan from part (c). For each of the following 5 changes to it, answer "Y" (yes) or "N" (no) depending on whether it could have led to a further reduction in the number of I/Os. Shortly explain your reasoning. Consider each change individually!

1. Pushing down the selection on `G.brand` and `G.price` below the join:

2. Creating a temporary file to store the results of the selection on `P.age`:

3. Have the index on `Guitars.gid` be clustered:

4. Projecting out `Players.age` and `LastPlayed.date` before the join.

5. Changing the first join to block-oriented nested loop join:

Initials: ____________________________

4. (**14** points) **Parallel Databases**

Consider the join $R(x, y) \bowtie S(y, z)$ between two relations, where R and S occupy 1000 and 300 blocks, respectively.

(a) (**4** points) What is the minimum cost in disk I/Os that is needed to join the relations using a block-nested loop join on a machine with 101 blocks of main memory?

(b) (**6** points) Now consider you want to execute the same query on a cluster of 4 machines, with each machine being identical to the one from part (a). Calculate the time needed in I/Os for the best parallel execution plan you can think of and describe this plan. Assume that tuples for both relations are equally distributed among the 4 machines in a round-robin fashion. Further assume that communication cost between the machines are negligible.

(c) (**4** points) Your classmate looks at what you have done and is completely fascinated: "Wow, you are so awesome! By using 4 instead of 1 machine, you have created a speed-up of more than factor 10. How on earth did you do that?" Be honest and explain the trick.

Initials: ____________________________

5. (**5** points + **5** extra credit) **Sorting**

A big company in Seattle called `amazort.com` decides to offer "sorting as a service." Users can upload a table of numbers, and the company will sort it for them and provide the sorted result for download. Unfortunately, after launching their advertising, the executives at `amazort.com` realize that they have no idea how to deal with large data. They hire you as a recent cse444 alumni to help them figure this out. The future of the company's reputation now relies on your shoulders.

(a) (**5** points) The new service sorts one file at a time. You promise the executives that the I/O count for sorting the file (including reading it from disk and writing the result to disk) will be 4 times that of simply reading it. To guarantee this, you suggest introducing a limit on the size of file uploads to some number `maxsize` bytes per file. The `amazort.com` servers use 64 kilobytes disk blocks, and have 512 megabytes of memory available for sorting. What value would you recommend for `maxsize` in kilobytes?

(b) (**5** extra credit) The chief executive of `amazort.com` is unhappy about the idea of size limits. "We are `amazort.com`, we have no limits." He asks you how many disk I/Os it would take to sort a file that is arbitrary x bytes big on a single server. At first, you are puzzled: the instructors of cse444 never taught you how to sort a file that is too big to allow for a two-pass algorithm, and you almost lose faith in your ability to save the future of `amazort.com`. After you calm down, you do what you are good at and why they hired you: you think. Can there be something like a 3-pass / 4-pass / n-pass sort algorithm that deals with increasingly large data? Yes, there is! You figure out that the second step in the two-pass algorithm simply generalizes, and with that, the formula for the maximal page size for a n-pass algorithm. Given file size x bytes, and M main memory blocks (M is sufficiently large to approximate $M - 1 \approx M$), how many I/Os are needed?

Initials: ____________________

EXTRA PAGE