Name Peter Cienporciento

Class Account: cs186-at

UNIVERSITY OF CALIFORNIA
Department of EECS, Computer Science Division

CS186 Hellerstein
Spring 2009 Midterm Exam #2

Midterm Exam: Introduction to Database Systems

This exam has 3 problems 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!*

1. SQL [15 points]

Consider the following schema:

Book (isbn, title, publisher, year) - primary key isbn - foreign key publisher references Publisher	**Publisher** (pname, city) - primary key pname
Author (ssn, name, birthplace) - primary key ssn	**Wrote** (isbn, ssn) - primary key isbn, ssn - foreign key isbn references Book - foreign key ssn references Author

Fill in the blanks of the following SQL queries.

a) **[5 points]** List the titles of all books written by the author named "David Foster Wallace".

```
SELECT b.title
FROM Book B, Author A, Wrote W
WHERE B.isbn = W.isbn
AND W.ssn = A.ssn
AND A.name = 'David Foster Wallace';
```

(Schema repeated from Page 1 for your convenience.)

Book *(isbn, title, publisher, year)* **Name** _____	***Publisher*** *(pname, city)*
- primary key isbn	*- primary key pname*
- foreign key publisher references Publisher	
	Wrote *(isbn, ssn)*
Author *(ssn, name, birthplace)*	*- primary key isbn, ssn*
- primary key ssn	*- foreign key isbn references Book*
	- foreign key ssn references Author

b.) [5 points] List the unique names of all authors who have written a book where one of the book's authors has the name Ramakrishnan (i.e., Ramakrishnan and all of his co-authors on all of his books). List results alphabetically by author's birthplace.

```
SELECT distinct CA.name
FROM Author a, Wrote w, Wrote CW, Author CA
WHERE A.name = 'Ramakrishnan'
AND A.ssn = W.ssn
AND W.isbn = CW.isbn
AND CW.ssn = CA.ssn
Order by CA.birthplace ASC
```

Name_____

(Schema repeated from Page 1 for your convenience.)

Book *(isbn, title, publisher, year)*	***Publisher*** *(pname, city)*
- primary key isbn	*- primary key pname*
- foreign key publisher references Publisher	
	Wrote *(isbn, ssn)*
Author *(ssn, name, birthplace)*	*- primary key isbn, ssn*
- primary key ssn	*- foreign key isbn references Book*
	- foreign key ssn references Author

c.) [5 points] List the names of **all** publishers in the city of San Francisco, and the number of books they published between the beginning of 2000 and the end of 2008.

SELECT P.name, count(*)

FROM Publisher P, Left Outer Join Book B

ON B.name = P.publisher

WHERE B.year between 2000 and 2008

AND P.city = 'San Francisco'

Group by ~~B.title~~
P.name

Name_____

(Schema repeated from Page 1 for your convenience.)

Book *(isbn, title, publisher, year)* *- primary key isbn* *- foreign key publisher references Publisher*	***Publisher*** *(pname, city)* *- primary key pname*
Author *(ssn, name, birthplace)* *- primary key ssn*	***Wrote*** *(isbn, ssn)* *- primary key isbn, ssn* *- foreign key isbn references Book* *- foreign key ssn references Author*

2. Relational Algebra and Calculus [18 points]

a.) [3 points] Our schema definition allows for authors who have never published a book. Write a relational algebra expression that returns only the names of authors that have published at least one book. Apply projections as early as possible.

$\Pi_{name}(\Pi_{ssn,name} A \bowtie \Pi_{ssn} W)$

b.) [3 points] Complete the relational calculus statement below to find titles of books written by Ferlinghetti that were published in cities other than San Francisco.

{T | ∃B ∃P ∃W ∃A (B ∈ Book ∧ P ∈ Publisher ∧ W ∈ Wrote ∧ A ∈ Author
∧ A.ssn = W.ssn

∧ W.isbn = B.isbn

∧ B.publisher = P.pname

∧ P.city != 'San Francisco'
∧ A.name = 'Ferlinghetti'

∧ T.title = B.title
)
}

Name_____

(Schema repeated from Page 1 for your convenience.)

Book *(isbn, title, publisher, year)* *- primary key isbn* *- foreign key publisher references Publisher*	***Publisher*** *(pname, city)* *- primary key pname*
Author *(ssn, name, birthplace)* *- primary key ssn*	***Wrote*** *(isbn, ssn)* *- primary key isbn, ssn* *- foreign key isbn references Book* *- foreign key ssn references Author*

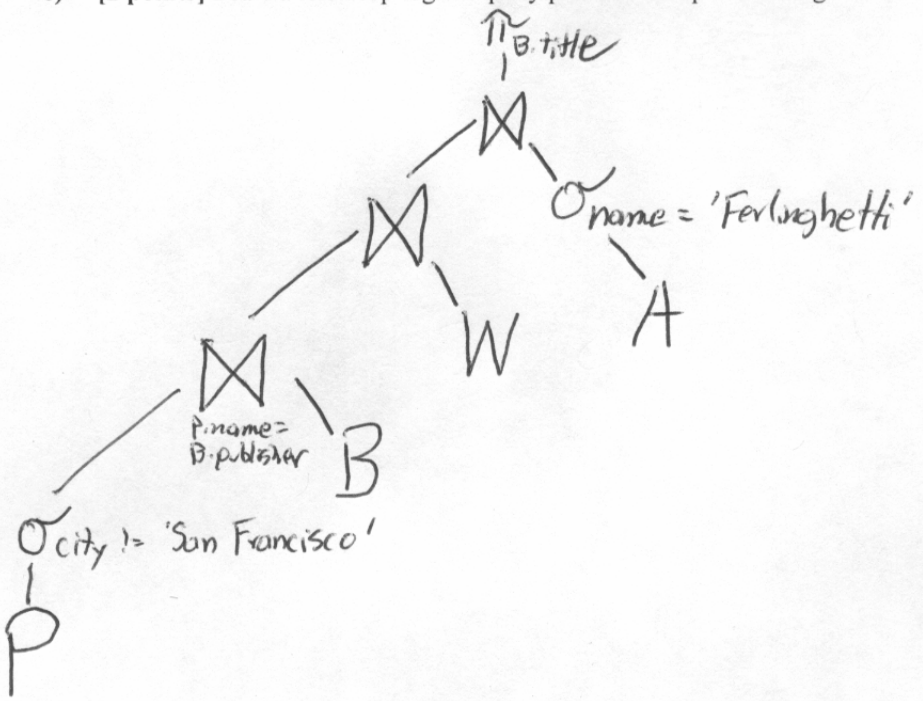
c) **[2 points]** Translate the Calculus formula from part (b) into a relational algebra statement. In this part, *leave all the selections and projections on the outside* (ie, apply selections and projectionsto the result of the joins, rather than pushing everything down. You may abbreviate the tables to their first letters.

$$\Pi_{B.title}\left(\sigma_{city != 'San\ Francisco'} \wedge A.name = 'Ferlinghetti' \left(P \underset{P.name = B.publisher}{\bowtie} B \bowtie W \bowtie A\right)\right)$$

d) **[2 points]** Now rewrite the expression of part (c), pushing selections down as far as possible.

$$\Pi_{B.title}\left(\sigma_{city != 'San\ Francisco'} P \underset{P.name = B.publisher}{\bowtie} B \bowtie W \bowtie \sigma_{name = "Ferlinghetti"} A\right)$$

e) **[2 points]** Draw a left-deep logical query plan for the optimized algebra statement of part (d).



Name_____

(schema repeated from page 1 for your convenience)

Book (isbn, title, publisher, year) - primary key isbn - foreign key publisher references Publisher	**Publisher** (pname, city) - primary key pname
Author (ssn, name, birthplace) - primary key ssn	**Wrote** (isbn, ssn) - primary key isbn, ssn - foreign key isbn references Book - foreign key ssn references Author

f) **[EXTRA CREDIT]** Ferlinghetti appears to publish mostly out of San Francisco. Let's call a "local author" a writer who has published only from a single city. Find all the local authors.

{A | A ∈ Author

∧ ∀ W ∃ P ∃B (W ∈ Wrote ∧ P ∈ Publisher ∧ B ∈ Book
∧ W.ssn = A.ssn

∧ W.isbn = B.isbn
∧ B.publisher = P.pname ⇒

¬ ∃ W2(W2 ∈ Wrote ∧ W2.ssn = W.ssn

∧ ∃ P2 ∃B2 (P2 ∈ Publisher

∧ B2 ∈ Book ∧ B2.ibsn = W2.ibsn
∧ B2.publisher = P2.pname

∧ P.city ≠ P2.city)

)

}

3. Query Optimization [18 Points]

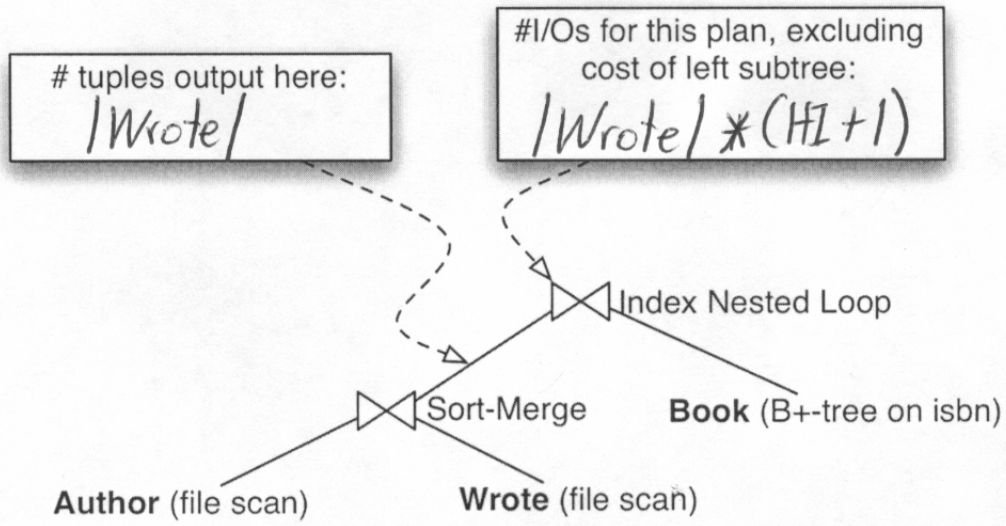
Useful notation and constants for this question (we offer shorthand in addition to the constant names discussed in class):

Notation	Meaning
|R| or NTuples	Cardinality of relation R
B_R or NPages	Number of disk pages occupied by R
TP_R	#Tuples per page of R
H_I or IHeight	Height of index I
IB_I or INPages	# of disk pages in index I
DE_I	# of Data Entries per leaf page of index I
#R.c or NKeys(R.c)	Number of distinct values in column c of relation R
Low(R.c)	Min value in column c of relation R
Hi(R.c)	Max value in column c of relation R
BSz	"block" ("batch") size for Block NL Join (in units of disk pages)
Bufs	Total number of frames in buffer pool

Consider the following query on the schema from Question 1:

```
SELECT *
  FROM Author NATURAL JOIN Wrote
              NATURAL JOIN Book;
```

a) Fill in the boxes in the query plan below with the appropriate expressions, using the notation from the table above. Note that the upper right box is *not* cumulative – you need not add in the cost of the lower left join, just the cost of the remainder of the plan. (6 Points)



Hint: when computing selectivities, do not simply apply the formulae from class. Think about everything you know about the tables involved.

Name_____

b) Consider the following query over the schema of Question 1:

```
SELECT * FROM BOOK
 WHERE isbn BETWEEN 10000 AND 10020;
```

Write an expression for the selectivity (reduction factor) of the WHERE clause in this query. (4 Points)

$$\frac{20}{Max(Book.isbn) - Min(Book.isbn)}$$

c) Consider the following query:

```
SELECT birthplace, COUNT(*) as cnt
  FROM Author NATURAL JOIN Wrote
              NATURAL JOIN Book
 GROUP BY birthplace
 ORDER BY cnt DESC;
```

Write down all the "interesting orders" at the start of query optimization. (4 Points)

Author.ssn, Wrote.ssn, Wrote.isbn, Book.isbn, Author.birthpla

d) During the query optimization algorithm, we enumerate plans for the subexpression (Author NATURAL JOIN Wrote). What "interesting orders" remain once that subexpression has already been optimized? (4 Points)

Wrote.isbn, Book.isbn, Author.birthplace

name: dummy
file: mt2
state: unknown