

## Midterm Exam 2: Introduction to Database Systems: Solutions

### 1. Query Optimization [12 points]

Consider the following schema.

auctions (aid, minprice, description, seller, end\_date).  
members (mid, nickname, name, since).  
bids (aid, buyerid, amount).

Assume there is an unclustered B-tree index on the key of each table. In answering questions, use the summary statistics functions that we learned about in class: NPages(), NTuples(), Low(), High(), NKeys(), IHeight(), INPages().

a. [3 points] Consider the query

```
SELECT 'found it!' FROM members WHERE mid = 98765;
```

Given the information above, write a formula for the optimizer's lowest estimated cost for this query.

IHeight(Btree on members.mid)

b. [3 points] Consider the query

```
SELECT * FROM bids, members
WHERE bids.buyerid = members.mid AND members.since < '2001';
```

Write the formula the optimizer would use for the selectivity of the entire WHERE clause.

[1/(MAX(Nvals(bids.buyerid), NVals(members.mid))

\*

[2001 - MIN(members.since)]  
/ [MAX(members.since)-MIN(members.since)]

Dcoetzee 4/19/10 11:50 PM
Deleted: ... Name ... [1]
Joe Hellerstein 4/4/10 5:01 PM
Formatted ... [2]
Joe Hellerstein 4/4/10 5:01 PM
Deleted: Student ID ... [3]
Dcoetzee 4/19/10 11:59 PM
Deleted: ... [4]

Dcoetzee 4/19/10 11:59 PM
Deleted: This exam has four problems ... [5]
Joe Hellerstein 4/3/10 10:49 PM
Deleted: 7
Joe Hellerstein 4/3/10 9:47 PM
Formatted: Bullets and Numbering ... [6]
Joe Hellerstein 4/3/10 10:17 PM
Deleted: an unclustered
Joe Hellerstein 4/3/10 10:09 PM
Deleted: <#> [3 points] Consider the ... [7]
Joe Hellerstein 4/3/10 9:47 PM
Formatted: Bullets and Numbering ... [8]
Joe Hellerstein 4/3/10 10:18 PM
Deleted: "
Joe Hellerstein 4/3/10 10:18 PM
Formatted ... [9]
Joe Hellerstein 4/3/10 10:18 PM
Deleted: "
Joseph Hellerstein 4/8/10 1:56 PM
Formatted ... [10]
Joseph Hellerstein 4/8/10 1:56 PM
Formatted ... [11]
Dcoetzee 4/19/10 11:59 PM
Deleted: ... [12]
Joe Hellerstein 4/3/10 9:47 PM
Formatted: Bullets and Numbering ... [13]
Joe Hellerstein 4/3/10 10:18 PM
Deleted: "
Joe Hellerstein 4/3/10 10:19 PM
Formatted ... [14]
Joe Hellerstein 4/3/10 10:19 PM
Formatted ... [15]
Joe Hellerstein 4/3/10 10:18 PM
Deleted: "HE
Joseph Hellerstein 4/8/10 1:58 PM
Formatted ... [16]
Joseph Hellerstein 4/8/10 1:57 PM
Formatted ... [17]
Dcoetzee 4/19/10 11:50 PM
Deleted: 2001

c. [2 points] Consider the same query as in part (b). Now suppose the optimizer knows that bids.buyerid is a “not null” foreign key referencing members. Write a simplified formula for the selectivity of the entire WHERE clause.

$1/NTuples(members)$

\*

$[2001 - MIN(members.since)]$   
 $/ [MAX(members.since) - MIN(members.since)]$

d. [4 points] Consider the following query:

```
SELECT R.*
FROM R, S, T
WHERE R.a = S.b
AND S.b = T.c
```

The following plans are generated during an intermediate pass of the Selinger (System R) optimizer algorithm. For each plan, write down the ordering column(s) of its output if any, and whether the plan would get pruned (P) or kept (K) at the end of the pass. If there is no clear ordering on the output, write “none”.

PLAN	Cost in I/Os	Ordering Columns of Output	Prune or Keep
IndexNestedLoops (FileScan(R), IndexOnlyScan(Btree on S.b))	2010	<u>None</u>	<u>P</u>
SortMergeJoin( FileScan(R), FileScan(S))	3010	<u>(R.a, S.b)</u>	<u>K</u>
ChunkNestedLoops( FileScan(R), FileScan(S))	1010	<u>None</u>	<u>K</u>
IndexNestedLoops( IndexOnlyScan(Btree on (R.d, R.a)), IndexScan(Btree on S.b))	30000	<u>(R.d, R.a)</u>	<u>P</u>

Dcoetzee 4/19/10 11:30 PM  
Deleted: ... Name ... [18]  
Joe Hellerstein 4/4/10 5:01 PM  
Formatted ... [19]  
Dcoetzee 4/20/10 12:00 AM  
Deleted: ... [20]  
Joe Hellerstein 4/3/10 9:47 PM  
Formatted: Bullets and Numbering ... [21]  
Joe Hellerstein 4/3/10 10:53 PM  
Deleted: 3...c ... [22]  
Dcoetzee 4/20/10 12:00 AM  
Deleted: ... [23]  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [24]  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [25]  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [26]  
Joseph Hellerstein 4/8/10 1:59 PM  
Formatted ... [27]  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [28]  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [29]  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [30]  
Joe Hellerstein 4/3/10 10:10 PM  
Deleted: ... [31]  
Dcoetzee 4/20/10 12:00 AM  
Deleted: ...  
Joseph Hellerstein 4/8/10 1:58 PM  
Formatted ... [32]  
Joe Hellerstein 4/3/10 10:49 PM  
Deleted: 5  
Joe Hellerstein 4/4/10 4:41 PM  
Formatted ... [33]  
Joe Hellerstein 4/4/10 4:41 PM  
Formatted ... [34]  
Joe Hellerstein 4/4/10 4:40 PM  
Formatted ... [35]  
Joe Hellerstein 4/4/10 4:41 PM  
Deleted: ... [36]  
Joseph Hellerstein 4/8/10 2:01 PM  
Formatted ... [37]  
Joseph Hellerstein 4/8/10 2:01 PM  
Formatted ... [38]  
Joseph Hellerstein 4/8/10 2:01 PM  
Formatted ... [39]  
Joseph Hellerstein 4/8/10 2:01 PM  
Formatted ... [40]  
Joseph Hellerstein 4/8/10 2:00 PM  
Deleted: S  
Joseph Hellerstein 4/8/10 2:00 PM  
Formatted ... [41]  
Joseph Hellerstein 4/8/10 2:01 PM  
Deleted: S.b  
Joseph Hellerstein 4/8/10 2:01 PM  
Formatted ... [42]  
Dcoetzee 4/19/10 11:50 PM

## 2. Relational Algebra/Calculus [16 points]

Consider the following schema:

Store(sid, store\_name, parent\_company).

Branch(sid, city, open24).

- sid is a foreign key to store
- open24 is a boolean. If true, the store is open 24 hours a day. If false, it is open 7AM-10PM.

Has\_Fruit(sid, city, fid, quantity, price).

- sid is a foreign key to store
- fid is a foreign key to fruit
- price is a floating point number representing the number of whole dollars and whole cents of the price (for example, 3.00)

Fruit(fruit\_id, name, calories).

- a. [3 points] Assume it is 2AM in Berkeley, you are craving some pineapple, and since you own stock in parent company "WFM", you want to buy your pineapple in a branch owned by them. Complete the relational calculus query below to return store names of all stores with branches in Berkeley that are open at 2AM, have at least 5 pineapples, charge less than \$3.00 per pineapple, and have parent company "WFM".

$\{ X \mid \exists S \exists B \exists H \exists F ( S \in \text{Store} \wedge B \in \text{Branch} \wedge H \in \text{Has\_Fruit} \wedge F \in \text{Fruit}$

$\wedge$

s.sid = b.sid AND  
s.sid = h.sid AND  
b.city = h.city AND  
h.fid = f.fruit\_id AND  
b.city = "berkeley" AND  
b.open24 AND  
f.name = "pineapple" AND  
h.quantity >= 5 AND  
h.price < 3.00 AND  
s.parent\_company = "wfm" AND  
x.store\_name = s.store\_name

}

*All 11 clauses correct; 3 points. 5 to 10 clauses; 2 points. 1 to 4 clauses; 1 point. Points were not deducted for switching </<= and >/>= or for missing quotes on strings or missing ".00" on floating-point values.*

Dcoetzee 4/19/10 11:50 PM

Deleted: Name

Joe Hellerstein 4/4/10 5:01 PM

Formatted: Font:Bold

Joe Hellerstein 4/3/10 10:50 PM

Deleted: 8

Joe Hellerstein 4/3/10 10:31 PM

Deleted: hour, close\_hour

Joe Hellerstein 4/3/10 10:31 PM

Deleted: hour and close\_hour are integers representing an hour of the day (in 24-hour time). For example, if a branch opens at 8AM and closes at 7PM, then open\_hour = 8 and close\_hour = 19

Joe Hellerstein 4/3/10 10:40 PM

Deleted: sids

Dcoetzee 4/22/10 2:54 PM

Deleted:

Dcoetzee 4/22/10 2:52 PM

Formatted: Font:Not Bold, Font color: Red

Dcoetzee 4/22/10 2:54 PM

Formatted: Indent: Left: 0.5"

Dcoetzee 4/22/10 2:52 PM

Formatted: Font:Not Bold, No underline, Font color: Red

Dcoetzee 4/22/10 2:51 PM

Deleted:

... [43]

Dcoetzee 4/22/10 2:52 PM

Formatted: Font color: Red

Dcoetzee 4/22/10 2:54 PM

Deleted:

Dcoetzee 4/22/10 2:55 PM

Formatted ... [44]

Dcoetzee 4/22/10 2:55 PM

Formatted ... [45]

Dcoetzee 4/22/10 2:55 PM

Formatted ... [46]

Dcoetzee 4/22/10 2:55 PM

Formatted ... [47]

Dcoetzee 4/22/10 2:55 PM

Formatted ... [48]

Dcoetzee 4/22/10 2:55 PM

Formatted ... [49]

Dcoetzee 4/22/10 2:55 PM

Formatted: Font:Italic, Font color: Red

Dcoetzee 4/19/10 11:50 PM

Deleted: 2001

- b. [4 points] Translate the query of part (a) to relational algebra, pushing selections in as far as possible, postponing projection until the very end, and using the join order: Store, Branch, Has\_Fruit, Fruit.

$\pi_{\text{store\_name}}(\sigma_{\text{parent\_company}='WFM'}(\text{Store}) \bowtie \sigma_{\text{city}='Berkeley' \text{ AND } \text{open24}}(\text{Branch})$   
 $\bowtie \sigma_{\text{quantity} \geq 5 \text{ AND } \text{price} < 3}(\text{Has\_Fruit}) \bowtie \sigma_{\text{fruit\_id}=\text{fid}, \text{name}='pineapple'}(\text{Fruit}))$

*1-2 points for correctness. If at least 1 point for correctness, then 1 point for pushing all selections down as far as possible, and 1 point for keeping all projections outside.*

*0 to 3 errors : 1 point off.  $\geq 4$  errors : 2 points off.*

*Points were deducted for:*

- using a join order resulting in a cross product
- missing a selection condition
- missing or invalid join conditions (e.g.  $\text{fid} = \text{fruit\_id}$ )
- missing tables in WHERE clause

- c. [5 points] Assume you are part of LowCal, a student group that encourages students to eat low-calorie fruit. Your group is starting a new campaign to boycott parent companies that have at least one store branch with no "Has\_Fruit" records for fruit with less than 300 calories. Complete the relational calculus expression below that lists the parent companies you should boycott.

$\{ X \mid \exists S \exists B ( S \in \text{Store} \wedge B \in \text{Branch} \wedge$   
 $\quad S.\text{sid} = B.\text{sid} \wedge X.\text{parent\_company} = S.\text{parent\_company},$   
 $\forall H \in \text{Has\_Fruit} ($   
 $\quad ( H.\text{sid} = B.\text{sid} \wedge H.\text{city} = B.\text{city} ) \Rightarrow$   
 $\quad \neg \exists F ( F \in \text{Fruit} \wedge F.\text{fruit\_id} = H.\text{fid} \wedge F.\text{calories} \geq 300 ) ) \}$

*0.5 points for having " $H.\text{sid} = B.\text{sid} \wedge H.\text{city} = B.\text{city}$ " in a context with correct quantifiers over H and B.*

*0.5 points for having " $X.\text{parent\_company} = S.\text{parent\_company}$ " in a context with correct quantifiers over X and S.*

*0.5 points for having " $S.\text{sid} = B.\text{sid}$ " in a context with correct quantifiers over S and B*

*0.5 point for having the equivalent of " $\exists F ( F \in \text{Fruit} \wedge F.\text{fruit\_id} = H.\text{fid} \wedge F.\text{calories} \geq 300 )$ " in a context with the correct quantifier over H.*

*3 points if the second part of your answer was logically equivalent to the solution above.*

Dcoetzee 4/19/10 11:50 PM

Deleted: Name

Joe Hellerstein 4/4/10 5:01 PM

Formatted: Font:Bold

Dcoetzee 4/22/10 3:52 PM

Formatted: Justified, Numbered + Level: 1 + Numbering Style: a, b, c, ... + Start at: 1 + Alignment: Left + Aligned at: 0" + Indent at: 0.25"

Dcoetzee 4/22/10 3:52 PM

Deleted:

Dcoetzee 4/22/10 3:01 PM

Formatted

... [50]

Dcoetzee 4/22/10 3:01 PM

Formatted

... [51]

Dcoetzee 4/22/10 3:01 PM

Deleted:

Dcoetzee 4/22/10 3:01 PM

Formatted: Font:Italic, Font color: Red

Joe Hellerstein 4/4/10 5:04 PM

Deleted: 4...sids of

... [52]

Dcoetzee 4/22/10 3:54 PM

Deleted:

Dcoetzee 4/22/10 3:52 PM

Formatted

... [53]

Dcoetzee 4/22/10 3:54 PM

Formatted: Indent: First line: 0.5"

Dcoetzee 4/22/10 3:54 PM

Deleted:

Joe Hellerstein 4/4/10 4:42 PM

Deleted:  $\exists F ( F \in \text{Fruit} \wedge$

Dcoetzee 4/22/10 3:54 PM

Deleted:

Dcoetzee 4/22/10 3:53 PM

Formatted

... [54]

Dcoetzee 4/22/10 3:53 PM

Formatted: Indent: Left: 0.5"

Dcoetzee 4/22/10 3:53 PM

Formatted

... [55]

Dcoetzee 4/22/10 3:54 PM

Deleted:

Dcoetzee 4/22/10 3:54 PM

Deleted:

... [56]

Dcoetzee 4/22/10 3:54 PM

Formatted

... [57]

Dcoetzee 4/19/10 11:50 PM

Deleted: 2001

- d. [4 points] Translate the expression [from part \(c\)](#) into a single SQL query (with a single “NOT EXISTS” sub-query) by completing the query below. Recall that SQL returns duplicates by default, whereas the relational calculus does not.

```
SELECT
    DISTINCT S.parent_company
FROM Store S, Branch B
WHERE
    S.sid = B.sid
AND NOT EXISTS (
    SELECT *
    FROM Has_Fruit H, Fruit F
    WHERE
        H.fid = F.fruit_id
        AND H.sid = S.sid
        AND F.calories >= 300
        AND H.city = B.city
);
```

*The following six expressions should have been included:*

1. *DISTINCT S.parent\_company*
2. *S.sid = B.sid*
3. *H.fid = F.fruit\_id*
4. *H.sid = S.sid*
5. *F.calories >= 300*
6. *H.city = B.city*

*All 6 were included: 4 points*

*4 or 5: 3 points*

*2 to 3: 2 points*

*1: 1 point*

*0: 0 points*

Dcoetzee 4/19/10 11:50 PM	Deleted: ... Name ... [58]
Joe Hellerstein 4/4/10 5:01 PM	Formatted ... [59]
Dcoetzee 4/22/10 3:54 PM	Formatted ... [60]
Joe Hellerstein 4/3/10 10:12 PM	Deleted: above
Dcoetzee 4/22/10 3:57 PM	Deleted: ...
Dcoetzee 4/22/10 3:56 PM	Formatted ... [61]
Dcoetzee 4/22/10 3:57 PM	Deleted: ...
Dcoetzee 4/22/10 3:57 PM	Formatted ... [62]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [63]
Dcoetzee 4/22/10 3:57 PM	Deleted: ... [64]
Dcoetzee 4/22/10 3:57 PM	Formatted ... [65]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [66]
Dcoetzee 4/22/10 3:57 PM	Deleted: ... [67]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [68]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [69]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [70]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [71]
Dcoetzee 4/22/10 3:58 PM	Formatted ... [72]
Dcoetzee 4/22/10 3:57 PM	Deleted: ... [73]
Dcoetzee 4/22/10 4:00 PM	Formatted ... [74]
Dcoetzee 4/22/10 4:00 PM	Formatted ... [75]
Dcoetzee 4/22/10 4:00 PM	Formatted ... [76]
Joe Hellerstein 4/3/10 10:12 PM	Deleted: ... [77]
Dcoetzee 4/19/10 11:50 PM	Deleted: 2001

### 3. Parallel Query Processing [8 points]

- a. [4 points] Skew can be a problem in certain query processing techniques. In the table below, please describe the problems that skew can cause for *single-table* sorting or hashing. In the first column please address the single-node algorithms that “spill” to disk; in the second column please address parallel algorithms and assume that the data that arrives at each nodes fits in RAM. In cases where skew does not cause problems, simply write “No Problem”. Your answer should be very short, and fit within the box!

	SINGLE NODE, SPILLS TO DISK	PARALLEL, FITS IN RAM
SORTING	<u>No Problem</u>	<u>Uneven distribution of tuples across machines will lead to stragglers</u>
HASHING	<u>Uneven distribution of tuples across partitions will lead to unnecessary recursive partitioning</u>	<u>Uneven distribution of tuples across machines will lead to stragglers</u>

- b. [4 points] Consider the query

SELECT key, AVG(val) FROM T GROUP BY key;

Both key and val are integer fields, and our machines use 4-byte integers and floats. Assume there are 10,000 key values, and 101 machines. Tuples are distributed randomly across all 101 machines to begin, and all 101 machines share the work of processing each query operator. How many bytes should this query send across the network, not including output? Briefly and neatly justify your answer by describing the data sent by each machine (your answer should fit easily below!)

<u>Transvals have form (key,count,sum)</u>	<u>12</u>	<u>Bytes/Transval</u>
<u># Transvals computed per node</u>	<u>10,000</u>	<u>Transvals/Node</u>
<u>Fraction of Transvals shipped</u>	<u>100/101</u>	<u>Nodes/Nodes</u>
<u># nodes sending (partitioned II-ism)</u>	<u>101</u>	<u>Nodes</u>
	<u>12,000,000</u>	<u>Bytes</u>

Dcoetzee 4/19/10 11:50 PM

Deleted: Name

Joe Hellerstein 4/4/10 5:01 PM

Formatted: Font:Bold

Joe Hellerstein 4/3/10 10:50 PM

Deleted: 16

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font:(Default) +Theme Body, No underline, Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font:(Default) +Theme Body, No underline, Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font:(Default) +Theme Body, No underline, Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font:(Default) +Theme Body, No underline, Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font color: Red

Joseph Hellerstein 4/8/10 2:03 PM

Formatted: Font color: Red

Joe Hellerstein 4/3/10 9:50 PM

Formatted: Font:(Default) Courier New, 11 pt, No underline, Font color: Auto

Joe Hellerstein 4/3/10 9:53 PM

Deleted: 0

Joe Hellerstein 4/3/10 9:52 PM

Deleted: , and that t

Joe Hellerstein 4/3/10 10:13 PM

Formatted: Font:(Default) +Theme Body, Italic, No underline, Font color: Auto

Joe Hellerstein 4/3/10 10:13 PM

Deleted: .

Joseph Hellerstein 4/8/10 3:40 PM

Deleted: .

Joseph Hellerstein 4/8/10 3:17 PM

Formatted Table

Joseph Hellerstein 4/8/10 3:01 PM

Formatted: List Paragraph, Indent: Left: 0.25"

Dcoetzee 4/19/10 11:50 PM

Deleted: 2001

#### 4. SQL [13 points]

Consider the table created by the following SQL statement:

```
CREATE TABLE G (    -- G is short for Grades
    ssn INT,          -- ssn of student
    class CHAR(5),    -- department and number of course
    grade INT,        -- final grade between 0 and 100
    PRIMARY KEY(ssn, class)
)
```

Fill in the blanks to produce valid SQL queries to answer the following questions.

- a. [3 points] Find the SSNs of students who received the maximum score in CS186 without using MAX.

```
SELECT ssn FROM G
```

```
WHERE class='CS186' AND grade >= ALL
```

```
(SELECT grade FROM G) ;
```

*One point was deducted for use of >=. No points were given for answers involving "100", since it was noted on the board during the exam that "maximum" refers to the highest score in the class, not the maximum possible score.*

- b. [3 points] Find the SSNs of all students who have received grades of 85 or above in at least 3 classes.

```
SELECT ssn FROM G
```

```
WHERE grade >= 85
```

```
GROUP BY ssn
```

```
HAVING COUNT(*) >= 3 ;
```

*Any valid expression was accepted inside the COUNT() aggregate. Using ">" instead of ">=" in one place was forgiven. Using "> 84" or "> 2" is okay. Using > in both the WHERE and HAVING clause, using the wrong GROUP BY, or reversing the WHERE and*

Dcoetzee 4/19/10 11:50 PM

Deleted: Name

Joe Hellerstein 4/4/10 5:01 PM

Formatted: Font:Bold

Joe Hellerstein 4/3/10 10:14 PM

Deleted:

Joe Hellerstein 4/4/10 4:39 PM

Deleted:

[4 points] Google's MapReduce implementation is able to recover from the failure of machines in the middle of a "job" without restarting everything from scratch. Your friend suggests using the lessons from Google as inspiration to recover during single-machine sorting, when you detect a failed disk block on the disk that you "spilled" sorted runs to. (Yes, this does happen, and can be detected by software!) Suppose that during the merge phase

Joe Hellerstein 4/4/10 4:39 PM

Deleted: Writing

Joe Hellerstein 4/4/10 4:39 PM

Deleted: queries

Dcoetzee 4/20/10 12:22 AM

Formatted: ... [79]

Dcoetzee 4/20/10 12:21 AM

Deleted: ... [80]

Dcoetzee 4/20/10 12:23 AM

Formatted: ... [81]

Dcoetzee 4/20/10 12:23 AM

Formatted: ... [82]

Dcoetzee 4/20/10 12:23 AM

Formatted: Font:Italic, Font color: Red

Dcoetzee 4/20/10 12:24 AM

Deleted: ... [83]

Dcoetzee 4/20/10 12:24 AM

Formatted: Underline, Font color: Red

Dcoetzee 4/20/10 12:24 AM

Deleted: ... [84]

Dcoetzee 4/20/10 12:24 AM

Formatted: Underline, Font color: Red

Dcoetzee 4/20/10 12:24 AM

Deleted: ... [85]

Dcoetzee 4/20/10 12:24 AM

Formatted: ... [86]

Dcoetzee 4/20/10 12:27 AM

Formatted: Standard

Dcoetzee 4/19/10 11:50 PM

Deleted: 2001

*HAVING clauses resulted in lost points.*

Dcoetzee 4/19/10 11:50 PM

Deleted: Name

Joe Hellerstein 4/4/10 5:01 PM

Formatted: Font:Bold

Dcoetzee 4/20/10 12:27 AM

Deleted:

Dcoetzee 4/20/10 12:25 AM

Formatted: Font:Italic, Font color: Red

Dcoetzee 4/19/10 11:50 PM

Deleted: 2001



- c. [3 points] For each CS186 student, return three columns: their SSN, a boolean value indicating whether they have taken CS161, and the average of their grades in the two classes combined. If they have not taken CS161, produce NULL for their average grade.

```
CREATE VIEW T1 AS
  SELECT ssn, grade FROM g WHERE class='CS186';
CREATE VIEW T2 AS
  SELECT ssn, grade FROM g WHERE class='CS161';
```

```
SELECT T1.ssn, T2.ssn IS NOT NULL, (T1.grade + T2.grade)/2,
FROM T1 LEFT OUTER JOIN T2
ON T1.ssn = T2.ssn ;
```

*Points were not deducted for correct use of CASE, although this was unnecessary (only for CASE with wrong syntax or result). Points were lost for the syntax "ON ssn" (invalid ambiguous reference to ssn), for the syntax "T2.ssn <> NULL" (this is NULL when T2.ssn is NULL, not false), for the use of aggregates such as AVG (aggregates cannot aggregate over columns, only over rows, and there is no GROUP BY here), for the wrong type of JOIN, and for many other reasons.*

- d. [4 points] Find pairs of distinct students who took at least 3 classes together. Each unordered pair should be listed only once.

```
SELECT G1.ssn, G2.ssn
FROM FROM G AS G1, G AS G2
WHERE WHERE G1.ssn < G2.ssn AND G1.class = G2.class
GROUP BY G1.ssn, G2.ssn
HAVING COUNT(*) >= 3 ;
```

*Two points were deducted for not including G1.ssn < G2.ssn or G1.ssn > G2.ssn – this was the essential condition needed to ensure that duplicate unordered pairs do not occur (e.g. (2,4) and (4,2)). Alternate table names instead of G1, G2 were accepted, as were redundant conditions such as G1.ssn <> G2.ssn; and you could put anything valid inside the COUNT() aggregate. Points were also deducted for excluding the condition G1.class = G2.class, for selecting from only one copy of G, for grouping by class instead of ssn, for placing the HAVING condition in the WHERE clause, for not including both ssn columns in the output, and for many other reasons.*

Deleted: ... Name	... [87]
Joe Hellerstein 4/4/10 5:01 PM	
Formatted	... [88]
Joe Hellerstein 4/3/10 10:06 PM	
Deleted: CS186 grades and CS161 grades	
Dcoetzee 4/20/10 12:28 AM	
Deleted:	... [89]
Dcoetzee 4/20/10 12:28 AM	
Deleted:	... [90]
Dcoetzee 4/20/10 12:28 AM	
Formatted	... [91]
Dcoetzee 4/20/10 12:28 AM	
Formatted	... [92]
Joe Hellerstein 4/3/10 10:01 PM	
Deleted: (SELECT ssn, grade	... [93]
Joe Hellerstein 4/3/10 10:02 PM	
Deleted:	... [94]
Dcoetzee 4/20/10 12:28 AM	
Formatted	... [95]
Dcoetzee 4/20/10 12:28 AM	
Formatted	... [96]
Dcoetzee 4/20/10 12:28 AM	
Deleted:	... [97]
Joe Hellerstein 4/3/10 10:02 PM	
Deleted:	... [98]
Dcoetzee 4/20/10 12:35 AM	
Deleted:	... [99]
Dcoetzee 4/20/10 12:30 AM	
Deleted:	
Dcoetzee 4/20/10 12:30 AM	
Formatted	... [100]
Joe Hellerstein 4/3/10 10:07 PM	
Deleted:	
Dcoetzee 4/20/10 12:36 AM	
Formatted	... [101]
Dcoetzee 4/20/10 12:35 AM	
Deleted:	... [102]
Dcoetzee 4/20/10 12:36 AM	
Deleted:	... [103]
Dcoetzee 4/20/10 12:36 AM	
Formatted	... [104]
Dcoetzee 4/20/10 12:36 AM	
Deleted:	... [105]
Dcoetzee 4/20/10 12:36 AM	
Formatted	... [106]
Dcoetzee 4/20/10 12:37 AM	
Deleted:	... [107]
Dcoetzee 4/20/10 12:37 AM	
Deleted:	... [108]
Dcoetzee 4/20/10 12:39 AM	
Deleted:	... [109]
Dcoetzee 4/20/10 12:39 AM	
Formatted	... [110]
Dcoetzee 4/19/10 11:50 PM	
Deleted: 2001	