Class Account: _____

UNIVERSITY OF CALIFORNIA
Department of EECS, Computer Science Division

CS186
Bohannon/Cooper
Spring 2008

First Exam: Introduction to Database Systems
February 26, 2008

Instructions:

1. Write your name on each page.
2. Turn in your notes page with your test.
3. There are 100 points total.
4. Please read over the test and plan your time. Best to skip and go back if you are stuck on a question.

Entity Relationship Graphs (14 pts)

You have just been hired as a consultant for a big airplane manufacturer. Impressed by your background in databases, they want you to completely redesign their database system. Talking with the people in the company, you get the following information.

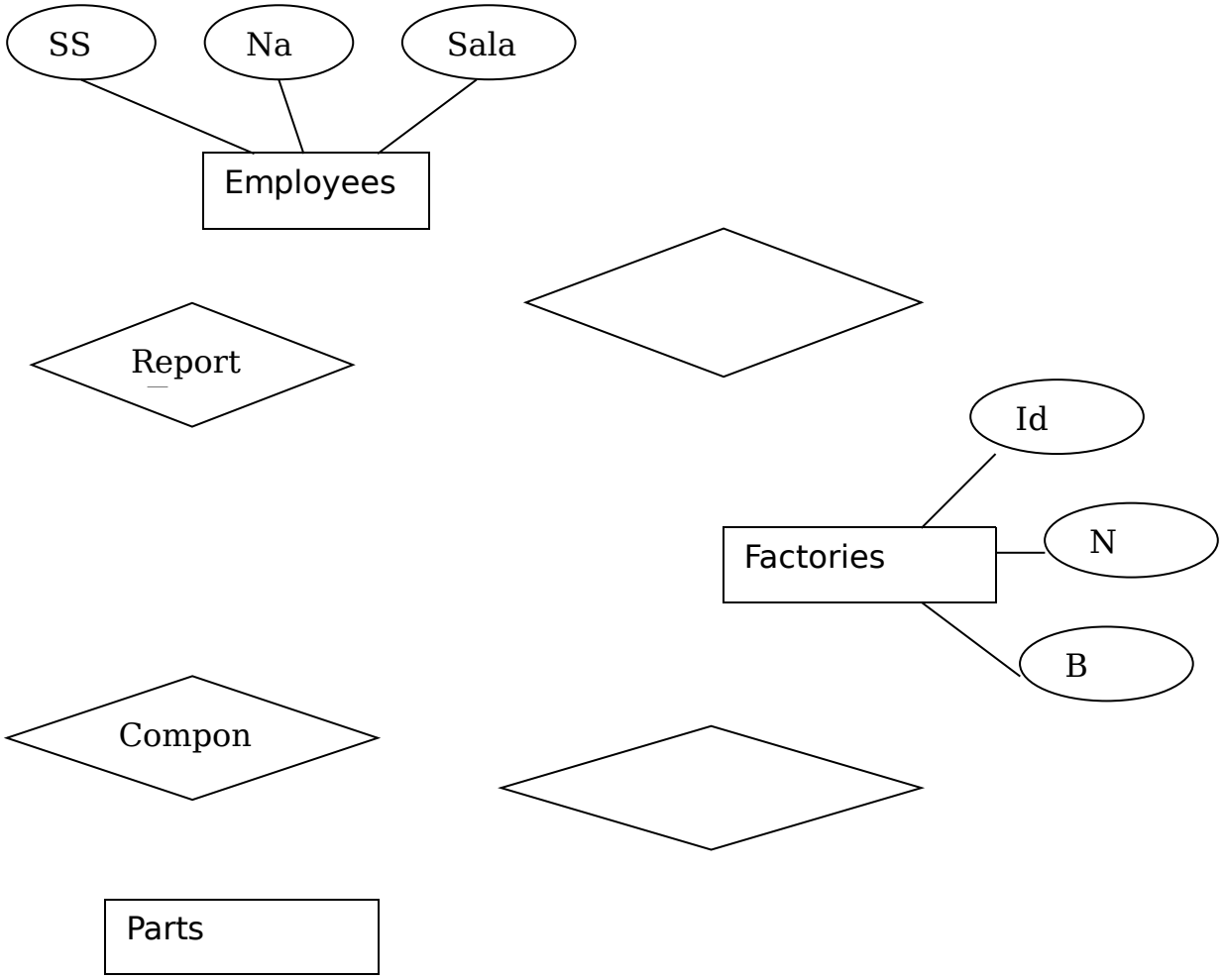
- The database contains information about employees, factories and parts.
- Each employee has a social security number (SSN), name and salary. An employee is uniquely identified by his or her SSN.
- Each factory has an id, name and a budget. The id uniquely identifies a project.
- Each part has an id and a name. The id uniquely identifies a part.
- Each employee reports to at most one other employee.
- Each employee works in at least one factory.
- Each part is manufactured in exactly one factory. Each part is a component of zero or more other parts.

A partial Entity-Relationship diagram for the above application is shown on the next page.

- add attributes of **parts**

- add missing relationships
- add keys of entity sets
- capture key and participation constraints on relationships

 Name:_____



File Organization and Structure (12 pts)

a) (6 pts) Consider a large table of students. You need to build an index for the table that best suits a given workload. Recall the 3 data entry alternatives for implementing an index, and the type of data structures that we have available

1. If 95% of the workload is a query requesting several pieces of information about students whose last name begins with a particular letter, and 5% of the workload is adding a new student, but the additions only occur on Saturday. what storage option is appropriate for ‘Students’.

2. If the percentage of the workload distribution is reversed and additions could occur any time, would the storage option change? Why?

b) (6 pts) Operating systems provide default file system management. Name two features that DBMS systems require that are not provided by

the default OS mechanisms, and specify whether they are provided at the file-layer or record-layer.

B+ Trees (12 pts)

This is an initial state of an index represented using a B+ tree. In this tree, under-full nodes are refilled from adjacent nodes if possible, otherwise a merge is done. Draw the B+ tree after the following operations have been completed. (Only the final B+ tree is needed but intermediate trees are needed for partial credit). Also, there is no need to draw the entire tree, you need to only redraw the nodes that are affected (label unchanged nodes with same N*x* labels used in the picture.)

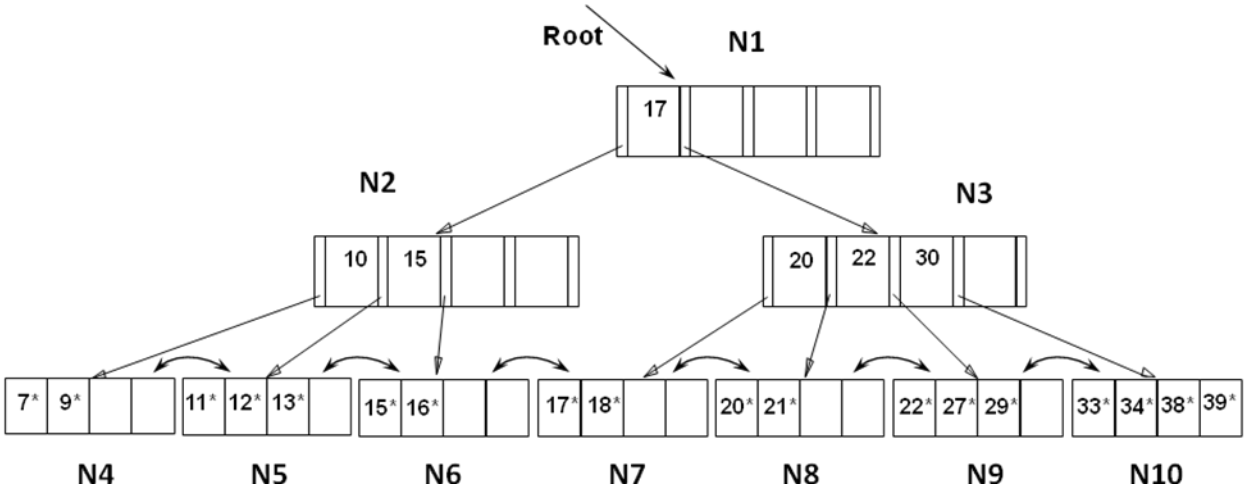
a) Insert record 40*

b) Delete record 7*

c) Delete record 9*

d) Delete record 15*

e) Delete record 18*



 Name:_____

(space for drawing B+Trees)

Sorting (14 pts)

Answer the following questions about external merge sorts.

a. (3 pts) Consider an external sort of a large file that does not fit in memory. What is the main impact of increasing the amount of memory available to the sort process?

i. The sort will likely proceed faster because the sort will read less data on each pass.
ii. The sort will likely proceed faster because the sort will potentially do fewer passes.
iii. The speed of the sort will not be affected.
iv. The sort will likely proceed slower because of the need to read more data on each pass.

ANSWER: _____

b. (3 pts) The primary benefit of double buffering is:

i. The ability to sort two files at once.
ii. The ability to merge two runs at once.
iii. The ability to reduce the number of passes in the sort.
iv. The ability to overlap computation and I/O.

ANSWER: _____

c. (4 pts) After each pass (except pass 0) of a 2-way external merge sort, we have:

i. Half as many runs of data as before the pass, but each run is twice as large
ii. Twice as many runs of data as before the pass, and each run is the same size
iii. The same number of runs of data as before the pass, and each run is the same size
iv. Half as many runs of data as before the pass, and each run is half as large

ANSWER: _____

d. (4 pts) Consider a system with 1 kilobyte pages. If we devote 1 gigabyte of memory space for input buffers, and we do not use double buffering, what is the largest file we can sort in two passes? (Remember 1 KB = 2^{10} bytes, 1 MB = 2^{20} bytes, 1 GB = 2^{30} bytes, 1 TB = 2^{40} bytes, and 1 PB = 2^{50} bytes)

i. 1 GB
ii. 100 GB
iii. 1 TB
iv. 1 PB

ANSWER: _____

Relational Algebra

Consider the following relations for a sales database for a chain of hardware stores:

Products

PRODUCT_ID	*NAME*	*MSRP*	*CATEGORY*	*NOTES*
42341	Drill	$35	Power tools	Cordless
43121	Hammer	$15	Hand tools	None
63433	Drywall nail	$0.15	Nails	2”
64221	Roofing nail	$0.15	Nails	1”
53433	Table saw	$215	Power tools	Safety shield

Stores

STORE_ID	*ADDRESS*	*REGION*	*MANAGER*
13	1 Main St., San Jose	WEST	Bob Smith
15	2 Elm Rd., NY	EAST	Naomi Smith
19	5 Shady Ln, Austin	CENTRAL	Tim Gunn
22	8 River Rd., Seattle	WEST	Johnny Rocket

Sales

DATE	*PRODUCT_ID*	*STORE_ID*	*QUANTITY*	*PRICE*
Jan. 1	43121	15	1	$15
Dec. 8	64221	22	100	$0.11
Mar. 3	53433	22	1	$210

Aug. 11	64221	13	1000	$0.15
Nov. 2	63433	15	50	$0.13
June 29	43121	15	2	$12
July 4	43121	15	1	$12

For the following queries, we have supplied some of the values for the results of the query. Fill in the missing values. Assume set semantics.

a $\pi_{NAME,QUANTITY}$ ($\sigma_{CATEGORY='Nails'}$ (Products ⋈ Sales)) (4 pts)

NAME	*QUANTITY*
Roofing nail	100
	1000
Drywall nail	

b. $\pi_{NAME,MANAGER,MSRP,PRICE}$ ($\sigma_{MSRP>PRICE}$(Products ⋈ Sales ⋈ Stores)) (5pts)

NAME	*MANAGER*	*MSRP*	*PRICE*
			$0.11
Table saw			$210
Hammer	Naomi Smith	$15	$12

c. $\pi_{MANAGER}$ ((π_{STORE_ID} (Stores) - π_{STORE_ID} (Sales)) ⋈ Stores) (4 pts)

MANAGER

d. (5 pts)

S1 = ρ (Sales1(DATE1, PRODUCT_ID1, STORE_ID1, QUANTITY1, PRICE1),Sales)
S2 = ρ (Sales2(DATE2, PRODUCT_ID2, STORE_ID2, QUANTITY2, PRICE2),Sales)
$\pi_{PRODUCTID1,PRICE1,PRICE2}$ ($\sigma_{DATE1 <> DATE2}$ ($\sigma_{PRODUCT\_ID1=PRODUCT\_ID2}$ S1 × S2))

PRODUCT_ID1	*PRICE1*	*PRICE2*
43121	$15	$12
64221	$0.11	$0.15

Relational Calculus

In this section, Assume there is an *additional table*, Manager, with a single attribute, NAME, and that there is foreign key constraint from Store.MANAGER to Manager.NAME.

a. (4 pts)

Write a (tuple) Relational Calculus expression to return Sales tuples indicating that the Shady Lane, Austin store sold more than 50 Hammers. Your answer can use the values shown above to get PRODUCT_ID and STORE_ID and avoid joins.

b. (4 pts)

Complete the following expression to return managers that only manage stores in the “WEST” region

M | M$\in$ Manager $\wedge$ $\forall$_____

c. (8 pts)
Write a Relational Calculus expression to find managers whose stores all either 1) sold more than 1000 of some (single) “Nail” product on some day or 2) sold a power tool costing over $100.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<eco>
<region name=”no cal”?>
    <species name=”rabbit” size=”small” native=”midwest”>
      <eats>
          <vegetable>carrot</vegetable>
          <vegetable>grass</vegetable>
     </eats>
      <endangered>never</endangered>
    </species>
    <species name=”fox” size=”small” native=”europe”>
      <eats><animal>mouse</animal><animal>rabbit</animal></eat
    s>
      <endangered>2000</endangered>
    </species>
</region>
<region name=”utah”> … </region>
<location name=”upper creek” region “no cal”>
    <plant>grass</plant>
    <plant>tree</plant>
    <plant>moss</plant>
</location>
<location> … </location>
</eco>
```

a. (4 pts)
Provide the declaration of the **eco** tag in a DTD:

eco → _____

b. (4 pts)
Write an XPath to give the name of species that live in the same region as a *strictly vegetarian* species. (Hint: only types of food are “animal” and “vegetable”).

c. (6 pts)
Complete the following XQuery that lists species that competes for food with an endangered species in a region.

```
<results>
{
for $r in doc/eco/region, $s  in $r/species
where $s/eats/animal =
$r/species_____
   or $s/eats/vegetable =
$r/species_____
 return { $s/@name }
}
</results>
```