Name

CWID

Quiz
1

Feburary 20th, 2013

Quiz 1: CS525 - Advanced Database Organization

Results

Please leave this empty! 1.1 1.2 1.3 1.4 Sum

Instructions

- Things that you are allowed to use
  - Textbook
  - Printed lecture notes
- Things that you are **not** allowed to use
  - Personal notes
- The quiz is **45** minutes long
- Start with the easy questions first to void getting stuck with a hard question and not finish the quiz.
- Multiple choice questions are graded in the following way: You get points for correct answers and points subtracted for wrong answers. The minimum points for each questions is **0**. For example, assume there is a multiple choice question with 6 answers - each may be correct or incorrect - and each answer gives 1 point. If you answer 3 questions correct and 3 incorrect you get 0 points. If you answer 4 questions correct and 2 incorrect you get 2 points. ...
- For your convenience the number of points for each part and questions are shown in parenthesis.
- There are 4 parts in this quiz
  1. SQL
  2. Relational Algebra
  3. Index Structures
  4. Result Size Estimation

Part 1.1 SQL (Total: 26 Points)

Consider the following database schema and instance:

user

name	limitMB
peter	50
hans	50
hilde	100
gertrud	1000
urs	1500

file

id	name	sizeMB	format
1	losung.pdf	1	PDF
2	urlaub.jpg	1	JPG
3	vorstellung.avi	347	Quicktime
4	stoiber.wmv	15	WindowsMediaVideo
5	faust.txt	2	Text
6	faust2.txt	1	Text
7	rambo3.avi	745	Quicktime

downloads

userName	fileId	date
peter	5	1.1.2007
peter	6	1.1.2007
hilde	7	25.7.2007
peter	2	1.1.2007
urs	1	4.9.2007

Hints:

- Underlined attribute form the primary key of an relation
- The attribut *userName* of relation *downloads* is a foreign key to *name* in relation *user*.
- The attribute *fileId* of relation *downloads* is a foreign key to attribute *id* in relation *file*.

Question 1.1.1 (3 Points)

Write an SQL statement that returns how much content (in MB) user "peter" has downloaded.

Solution

```
SELECT sum(d.sizeMB)
FROM   file d, downloads l
WHERE  d.id = l.fileID
       AND b.name = 'peter'
```

or

```
SELECT sum(d.sizeMB)
FROM file d, downloads l, user b
WHERE d.id = l.fileID
      AND b.name = 'peter'
      AND b.name = l.userName
```

Question 1.1.2 (3 Points)

Write an SQL statement that returns all files with id, name, and file size that were downloaded on 01/01/2007.

Solution

```
SELECT d.id, d.name
FROM file d, downloads l
WHERE d.id = l.fileId
      AND l.datum = '1.1.2007'
```

Question 1.1.3 (3 Points)

What is the result of evaluating the following SQL query?

```
SELECT b.name, d.name AS filename, d.sizeMB
FROM   file d, user b, downloads l
WHERE  b.name = l.userName
       AND d.id = l.fileId
       AND d.sizeMB > b.limitMB
```

Solution

name	filename	sizeMB
hilde	rambo3.avi	745

Question 1.1.4 (5 Points)

Write an SQL query that returns the total amount of data downloaded by each user.

Solution

```
SELECT sum(sizeMB), u.name
FROM user u, downloads d, file f
WHERE u.name = d.userName
      AND d.fileId = f.id
GROUP BY u.name
```

Question 1.1.5 (5 Points)

Write an SQL query that returns users (name, limitMB) which have exceeded their limit (have downloaded in total more then `limitMB` megabytes of files).

Solution

```
SELECT u.name, u.limitDB
FROM user u, downloads d, file f
WHERE u.name = d.userName
      AND d.fileId = f.id
GROUP BY u.name, u.limitMB
HAVING u.limitMB < sum(sizeMB)
```

Question 1.1.6 (7 Points)

Write an SQL query that returns pairs of users (their names) that have downloaded the same files (the sets of files downloaded by both users is the same).

Solution

```
SELECT u1.name, u2.name
FROM user u1, user u2
WHERE u1.name < u2.name
      AND NOT EXISTS (SELECT *
                      FROM
                          (SELECT userName AS user1, fileId
                          FROM downloads
                          WHERE userName = u1.name) d1
                          FULL OUTER JOIN
                          (SELECT userName AS user2, fileId
                          FROM downloads
                          WHERE userName = u2.name) d2
                          ON (d1.fileId = d2.fileId)
                      WHERE user1 IS NULL OR user2 IS NULL)
```

or

```
SELECT user1, user2
FROM
     (SELECT count(*) AS td, userName FROM downloads GROUP BY userName) AS u1,
     (SELECT count(*) AS td, userName FROM downloads GROUP BY userName) AS u2,
     (SELECT count(*) AS td, l.userName AS user1, r.userName AS user2
     FROM downloads l JOIN downloads r ON (r.fileId = l.fileId)
     WHERE l.userName < r.userName
     GROUP BY l.userName, r.userName) AS b
WHERE u1.userName = user1 AND u2.userName = user2 AND u1.td = u2.td AND u1.td = b.td
```

alternatively using `EXCEPT`

alternatively using two nested `NOT EXISTS`: There are **no** files downloaded by this user that **not** have been downloaded by the other user.

Part 1.2 Relational Algebra (Total: 20 Points)

Question 1.2.1 Relational Algebra (5 Points)

Write an relational algebra expression over the schema from the SQL part (part 1) that returns all users (the *name*) which have downloaded more than 300MB of PDF files (bag semantics).

Solution

$$q = \pi_{name}(\sigma_{sum(sizeMB)>300}({}_{sum(sizeMB)}\alpha_{name}(join)))$$
$$join = \textbf{user} \bowtie_{name=userName} \textbf{downloads} \bowtie_{fileId=id} \rho_{fname \leftarrow name}(\sigma_{format=PDF}(\textbf{file}))$$

Question 1.2.2 SQL → Relational Algebra (5 Points)

Translate the following SQL query into relational algebra (bag semantics).

```
SELECT b.name, d.name AS filename, d.sizeMB
FROM   file d, user b, downloads l
WHERE  b.name = l.userName
       AND d.id = l.fileId
       AND d.sizeMB > b.limitMB
```

Solution

$$q = \pi_{name,filename,sizeMB}(\sigma_{sizeMB>limitMB}(join))$$
$$join = \textbf{user} \bowtie_{name=userName} \textbf{downloads} \bowtie_{fileId=id} \rho_{filename \leftarrow name}(\textbf{file}))$$

Question 1.2.3 SQL → Relational Algebra (4 Points)

Translate the following SQL query into relational algebra (bag semantics).

```
SELECT sum(sizeMB) AS totlMB, format
FROM file
GROUP BY format
HAVING format = 'Text' OR format = 'Quicktime'
```

Solution

$$q = \rho_{totlMB \leftarrow sum(sizeMB)}(\sigma_{format=Text \vee format=Quicktime}({}_{sum(sizeMB)}\alpha_{format}(file)))$$

Question 1.2.4 Equivalences (6 Points)

Consider the following relational schemata:

$R(A, B)$, $S(B, C)$, $T(C, D)$.

Check the equivalences that are correct (set semantics). For example $R \equiv R$ should be checked, whereas $R \equiv S$ should not be checked.

- ■ $R \bowtie (S \bowtie T) \equiv (T \bowtie R) \bowtie S$
- ■ $\pi_{A,B}(R \rtimes_{B=C} T) \equiv \pi_{A,B}(R \bowtie_{B=C} T) \cup R$
- ■ $R - S \equiv R - (R \cap S)$
- ■ $R \bowtie R \equiv R$
- ❑ $(R \cup S) \bowtie (R \cup T) \equiv R \bowtie (S \cup T)$
- ■ $R \rtimes\ltimes_{B=C} T \equiv (R \rtimes_{B=C} T) \cup (R \bowtie\ltimes_{B=C} T)$
- ■ $R \cap S \equiv R - (R - S)$
- ❑ $\sigma_{B=3}(R) \rtimes_{B=C} T \equiv R \rtimes_{B=C} \sigma_{C=3}(T)$
- ■ $\sigma_{A=3}({}_A\alpha_{sum(B)}(R)) \equiv {}_A\alpha_{sum(B)}(\sigma_{A=3}(R))$
- ■ $R \triangleright \rho_{A \leftarrow B, B \leftarrow C}(S) \equiv R - S$
- ■ $R \cup (S - R) \equiv S \cup (R - S)$
- ❑ $R - (S \cup T) \equiv T - (R \cup S)$

Part 1.3 Index Structures (Total: 24 Points)

Assume that you have the following table:

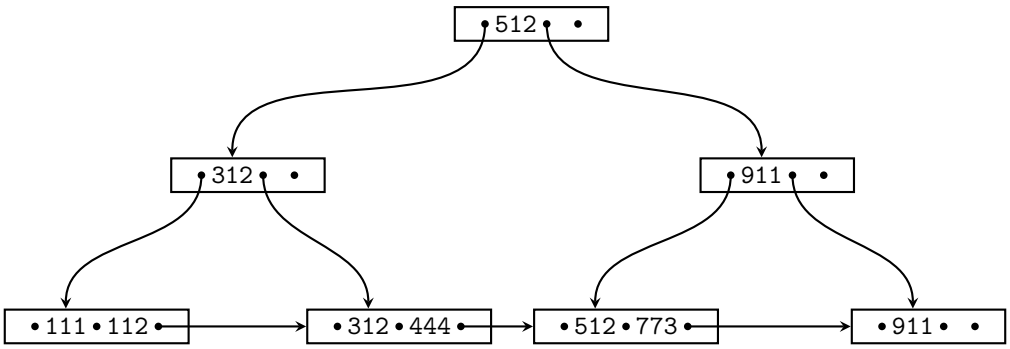
AreaZip

areacode	zip
312	60616
773	60616
512	50143
111	67777
911	-
112	11232
444	12322

Question 1.3.1 Construction (12 Points)

Create a B+-tree for table *AreaZip* on key *areacode* with $n = 2$ (up to two keys per node). Only write down the result of the construction.

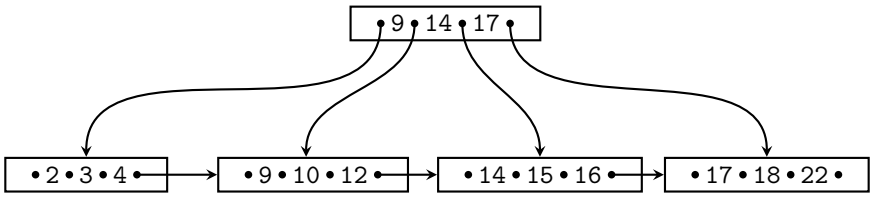
Solution



Question 1.3.2 Operations (12 Points)

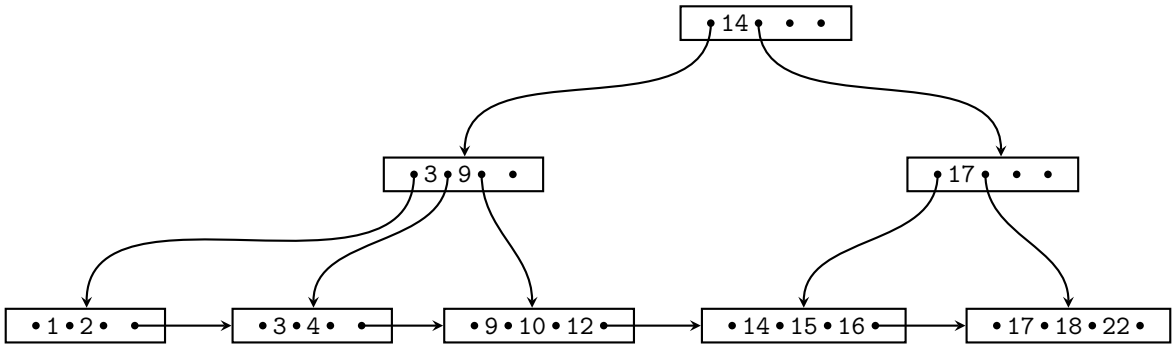
Given is the B+-tree shown below ($n = 3$). Execute the following operations and write down the resulting B+-tree:

insert(1), delete(14), delete(15), insert(20)

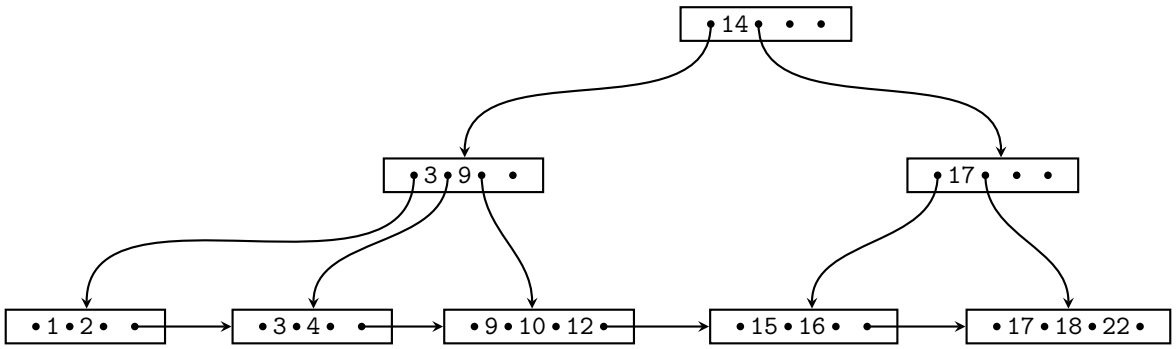


Solution

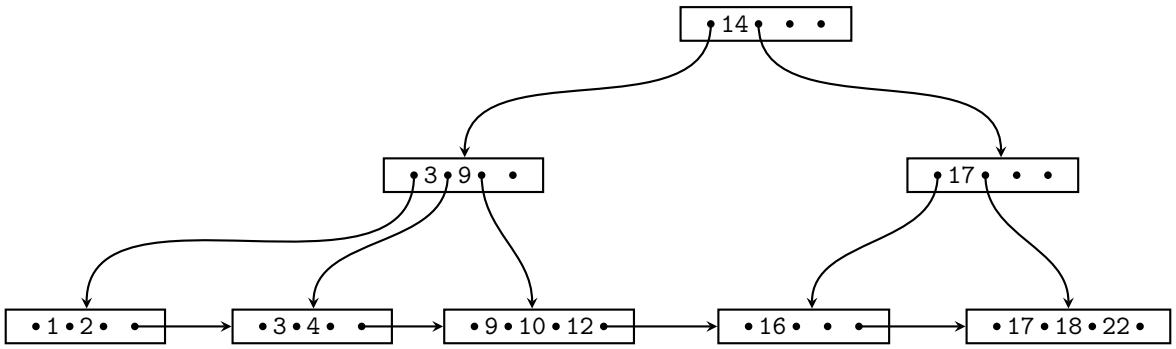
insert(1)



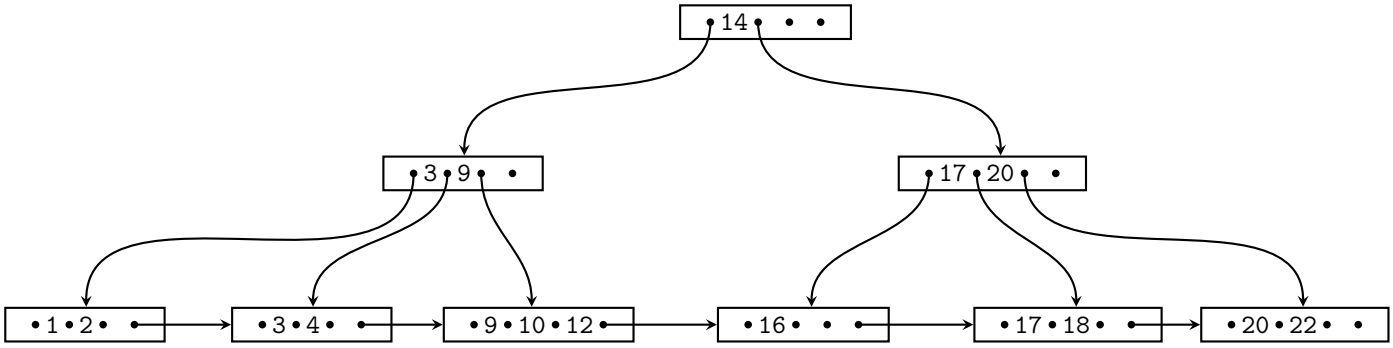
delete(14)



delete(15)



insert(20)



Part 1.4 Result Size Estimations (Total: 10 Points)

Consider a table *emp* with attributes *ssn*, *name*, *depId*, *salary*, a table *dep* with *id*, *department*, *city*, and a table *loc* with attributes *city* and *street*. *emp.depId* is a foreign key to department *depId*. Given are the following statistics:

$$T(emp) = 1000 \qquad T(dep) = 30 \qquad T(loc) = 30$$
$$V(emp, ssn) = 1000 \qquad V(dep, id) = 30 \qquad V(loc, city) = 10$$
$$V(emp, name) = 980 \qquad V(dep, department) = 30 \qquad V(loc, street) = 29$$
$$V(emp, depId) = 28 \qquad V(dep, city) = 10$$
$$V(emp, salary) = 120$$

Question 1.4.1 Estimate Result Size (2 Points)

Estimate the number of result tuples for the query $q = \sigma_{ssn=123}(emp)$ using the first assumption presented in class.

Solution

$$T(q) = \frac{T(emp)}{V(emp, ssn)} = \frac{1000}{1000} = 1$$

Question 1.4.2 Estimate Result Size (3 Points)

Estimate the number of result tuples for the query $q = emp \bowtie_{depId=id} dep$ using the first assumption presented in class.

Solution

$$T(q) = \frac{T(emp) \times T(dep)}{max(V(emp, depId), V(dep, depId))} = \frac{1000 \times 30}{max(28, 30)} = 1000$$

Question 1.4.3 Estimate Result Size (5 Points)

Estimate the number of result tuples for the query $q = emp \bowtie_{depId=id} dep \bowtie loc$ using the first assumption presented in class.

Solution

$$T(q) = \frac{T(emp) \times T(dep) \times T(loc)}{max(V(emp, depId), V(dep, depId)) \times max(V(dep, city), V(loc, city))} = \frac{1000 \times 30 \times 30}{max(28, 30) \times max(10, 10)} = 3000$$