Exercise Sheet 6

Question 13: Basic relational algebra

Given below are two tables $T_1$ and $T_2$.

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>x</td>
<td>15</td>
<td>e</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>x</td>
<td>10</td>
<td>f</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>x</td>
<td>15</td>
<td>g</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>y</td>
<td>15</td>
<td>h</td>
</tr>
</tbody>
</table>

$T_2$:

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>X</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>p</td>
<td>x</td>
<td>b</td>
</tr>
<tr>
<td>5</td>
<td>t</td>
<td>x</td>
<td>b</td>
</tr>
<tr>
<td>6</td>
<td>p</td>
<td>y</td>
<td>a</td>
</tr>
<tr>
<td>7</td>
<td>r</td>
<td>y</td>
<td>a</td>
</tr>
</tbody>
</table>

a. Calculate the result of the following relational algebra expression step by step:

$$\sigma_{D=p}(\pi_{X,Y}(T_1) \bowtie \pi_{X,D}(T_2))$$

b. Can you think of other ways of writing the above expression that produce the same result but involve less calculation?

Question 14: Understanding relational algebra

Explain each of the following relational algebra expressions in plain English, and translate them into SQL. (These refer to the “fundamentals” database.)

a. $\pi_{\text{lastname}}(\sigma_{\text{firstname}='John'}(\text{staff}))$

b. $\pi_{\text{lastname}}(\sigma_{\text{numbers}>100}(\text{staff} \bowtie \text{lecturing}))$

c. $\pi_{\text{name}}(\sigma_{\text{numbers}>100}(\text{lecturing} \bowtie \text{courses})) - \pi_{\text{name}}(\sigma_{\text{level}=1}(\text{courses}))$

d. $\pi_{\{\text{last name}, \text{name}\}}(\sigma_{\text{year}=1999}(\text{staff} \bowtie \text{lecturing} \bowtie \text{courses})) \cap \pi_{\{\text{last name}, \text{name}\}}(\sigma_{\text{level}=2}(\text{staff} \bowtie \text{lecturing} \bowtie \text{courses}))$

Question 15: From SQL to relational algebra

Translate the following SQL queries into relational algebra.

a. 

```sql
SELECT c.name
FROM courses AS c, staff AS s, lecturing AS l
WHERE l.sid = s.sid AND l.cid = c.cid AND
(l.year = 1999 OR l.year = 2000) AND s.lastname = 'Jung';
```

b. 

```sql
SELECT c.name
FROM lecturing AS l, courses AS c
WHERE l.cid = c.cid AND l.year = 2001
EXCEPT
SELECT c.name
FROM lecturing AS l, courses AS c
WHERE l.cid = c.cid AND (l.year = 2000 OR l.year = 1999);
```

c. 

```sql
SELECT c.name
FROM lecturing l1, lecturing l2, courses c
WHERE l1.cid=l2.cid AND l1.cid=c.cid AND
l1.sid=l2.sid AND
l1.year=1999 AND l2.year=2000;
```
d. SELECT s.lastname
    FROM staff s
    WHERE s.sid NOT IN (SELECT l.sid
                FROM lecturing l);

**Question 16: Deriving functional dependencies**

For each of the following assumed sets of functional dependencies (over the schema \((A, B, C, D)\)) determine whether they imply \(A \rightarrow D\). If yes, justify your answer with the closure algorithm; if no, give a table which satisfies the assumed dependencies but not \(A \rightarrow D\).

a. \(AB \rightarrow C, C \rightarrow B, B \rightarrow D\)
b. \(AB \rightarrow C, CB \rightarrow D, A \rightarrow B\)
c. \(ABC \rightarrow D, AB \rightarrow C, A \rightarrow B\)

**Question 17: Suggesting functional dependencies**

Consider the following schema which describes M.O.T. inspections of motor vehicles.

```
inspection(date_of_inspection,
    owner,
    owner_address,
    owner_contact_phone,
    registration_number,
    model,
    year_of_first_registration,
    diesel_or_petrol,
    date_of_previous_MOT,
    engineer,
    garage,
    garage_address,
    garage_MOT_licence_number,
    passed_or_not)
```

a. Find plausible (and non-trivial) functional dependencies. In doing so, list your assumptions and discuss whether they are reasonable.
b. Determine the candidate key(s).

d. **Question 18: Outer Joins (Optional)**

Given below are two tables \(T_1\) and \(T_2\).

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>12</td>
</tr>
</tbody>
</table>

a. Calculate the (natural) inner join \(T_1 \bowtie T_2\) of the two tables.
b. Which tuples of \(T_1\) and \(T_2\) are dangling tuples in this calculation?
c. Calculate the (natural) outer join of the two tables. (This is denoted \(T_1 \bowloid T_2\).)
d. Which tuples of the outer join are included if we are only interested in the *left* outer join?