# *Exercise Sheet 2*

## Question 1: Basic relational algebra

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

$T_1$ :

| V | W | X | Y | Z |
|---|---|---|---|---|
| 1 | $a$ | $x$ | 15 | $e$ |
| 2 | $a$ | $x$ | 10 | $f$ |
| 2 | $b$ | $x$ | 15 | $g$ |
| 2 | $b$ | $y$ | 15 | $h$ |

$T_2$ :

| C | D | X | W |
|---|---|---|---|
| 4 | $p$ | $x$ | $b$ |
| 5 | $t$ | $x$ | $b$ |
| 6 | $p$ | $y$ | $a$ |
| 7 | $r$ | $y$ | $a$ |

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 2: 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. $\pi_{\texttt{lastname}}(\sigma_{\texttt{firstname}='\texttt{John}'}(\texttt{staff}))$

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

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

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

## Question 3: From SQL to relational algebra

Translate the following SQL queries into relational algebra.

a.
```
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.
```
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.
```
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 4: 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).

## Question 5: Outer Joins (Optional))

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

$T_1$ :

| **A** | **B** | **C** |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

$T_2$ :

| **B** | **C** | **D** |
|---|---|---|
| 2 | 3 | 10 |
| 2 | 3 | 11 |
| 2 | 6 | 10 |
| 6 | 7 | 12 |

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 $T_1 \overset{\circ}{\bowtie} T_2$.)

d. Which tuples of the outer join are included if we are only interested in the *left* outer join?