CS 2541: In-Class Exercises. Relational Algebra and Relational Calculus

- **lives**(person-name, street, city)
- **works**(person-name, company-name, salary)
- **located-in**(company-name, city)
- **manages**(person-name, manager-name)

For the above schema (the primary key for each relation is denoted by the underlined attribute), provide relational algebra expressions for the following queries:

**Note:** For notational convenience, I am using `pname` instead of person-name, `cname` instead of company-name, and `mname` instead of manager-name. The SQL queries provided here are 'translation' of the relational algebra (RA) queries - there may be better/efficient ways to write them.

1. Find all tuples of persons who work for City Bank.
   
   \[
   \sigma_{\text{cname} = \text{\textquoteleft City Bank\textquoteright}}(\text{works})
   \]
   
   In relational calculus: \{\text{w} | \text{w} \in \text{works} \land (\text{w[cname]} \,=\, \text{\textquoteleft City Bank\textquoteright})\}\n
2. Find the name of all employees (i.e., persons) who work for the City Bank company (which is a specific company in the database).
   
   \[
   \pi_{\text{person-name}}(\sigma_{\text{cname} = \text{\textquoteleft City Bank\textquoteright}}(\text{works}))
   \]
   
   Relational calculus query. Note that \text{p} is defined only for person name attribute as per the right hand side of the query.
   
   \{\text{p} | \exists \text{w} \in \text{works}(\text{p[pname]} = \text{w[pname]} \land (\text{w[cname]} = \text{\textquoteleft City Bank\textquoteright}))\}\n
3. Find the name of all employees (i.e., persons) who work for the City Bank company and earn more than 50,000. This is similar to previous query with one additional predicate (of salary greater than 50000).
   
   \[
   \pi_{\text{person-name}}(\sigma_{\text{cname} = \text{\textquoteleft City Bank\textquoteright}}(\text{works}) \land (\text{salary} > 50000))
   \]
   
   Relational calculus query. Note that \text{p} is defined only for person name attribute as per the right hand side of the query.
   
   \{\text{p} | \exists \text{w} \in \text{works}(\text{p[pname]} = \text{w[pname]} \land (\text{w[cname]} = \text{\textquoteleft City Bank\textquoteright}) \land (\text{w[salary]} > 50000))\}\n
4. Find the name and city of all employees who work for City Bank and earn more than 50,000. Similar to previous query, except we have to access the lives table to extract the city of the employee. The join condition is the same person name in the two tables Lives and Works.

   \[
   \pi_{\text{lives.pname,lives.city}}(\sigma((\text{cname} = \text{\textquoteleft City Bank\textquoteright}) \land (\text{salary} > 50000)) \land (\text{lives.pname} = \text{works.pname}))(\text{works} \times \text{lives})
   \]

   Relational calculus query, note that the 'type' of \text{p} has two attributes - name and city- as referenced in the right hand side of the query.
\{p \mid \exists w \in \text{works, } \exists l \in \text{lives,} \}
\begin{align*}
&((w[cname] = 'City Bank') \land (w[salary] > 50000) \land (w[pname] = l[pname]) \\
&\land (l[pname] = p[pname] \land (w[city] = p[city])))\}
\end{align*}

In SQL:
SELECT lives.pname, city
FROM works, lives
WHERE cname='City Bank' AND salary> 50000 AND works.pname=lives.pname;

Note: In relational calculus, the notation \(x[name]\) is equivalent to \(x.name\) if you prefer to use syntax similar to SQL (or Rel algebra) to refer to attributes.

5. Find all employees who live in the same city as the company they work for. For this query we need to access the lives table to get city of the employee and the located-in table to get city of the company; plus the works table to associate employee with their company. The selection condition is then that the two cities are the same.

\[\pi_{\text{lives.pname}}(\sigma((\text{locatedin.cname} = \text{works.cname}) \land (\text{located-in.city} = \text{lives.city})) \land (\text{lives.pname} = \text{works.pname}))(\text{works} \times \text{lives} \times \text{locatedin}))\]

Relational calculus query

\{p \mid \exists x \in \text{lives, } \exists y \in \text{works, } \exists z \in \text{locatedin,} \}
\begin{align*}
&((p.pname = x.pname) \land (x.city = z.city) \land \\
&(x.pname = y.pname) \land (y.cname = z.cname))\}
\end{align*}

In SQL:
SELECT lives.pname
FROM lives, works, locatedin
WHERE lives.pname=works.pname AND works.cname=locatedin.cname
AND lives.city = locatedin.city;

6. Find all employees who live in the same city and on the same street as their manager. This requires accessing lives table twice – once for finding city of employee and a second time for finding city of manager. Therefore we need the rename operator – so access and rename lives as mlives to indicate this is for information about where the manager lives. To find manager of employee we need to look up the manages table.
\[ \pi_{\text{lives.pname}} \]
\[ (\sigma((\text{lives.city}=\text{mlives.city}) \land (\text{lives.street}=\text{mlives.street}) \land (\text{manages.pname}=\text{lives.pname}) \land (\text{mname}=\text{mlives.pname})) \]
\[ (\text{lives} \times \text{manages} \times (\rho_{\text{mlives(lives)}}))) \]

\[ \{ x \mid \exists y, z \in \text{lives} \exists m \in \text{manages} \]
\[ ((z.\text{city} = y.\text{city}) \land (y.\text{street} = z.\text{street}) \land
\]
\[ (y.\text{pname} = m.\text{pname}) \land (z.\text{pname} = m.\text{managername}) \land (x.\text{pname} = y.\text{pname})) \} \]

In SQL:

```
SELECT E.pname  
FROM lives E, lives M, manages  
WHERE E.pname= manages.pname AND manages.mname= M.pname  
   AND E.city= M.city AND E.street=M.street;
```