Exercise 1

Consider the following relational database:

employee(employee_name, street, city)
works(employee_name, company_name, salary)
company(company_name, city)
manages(employee_name, manager_name)

For each of the following queries, give an expression in relational algebra and SQL.

1. Find the names of all employees who work for First Bank Corporation.

\[ \pi \text{employee-name} (\sigma \text{company-name} = \text{"First Bank Corporation"} (\text{works})) \]

```sql
select employee_name
from works
where company name = 'First Bank Corporation';
```

2. Find the names and cities of residence of all employees who work for First Bank Corporation.

\[ \pi \text{employee-name, city} (\text{employee} \mid \text{works}) (\sigma \text{company-name} = \text{"First Bank Corporation"} (\text{works})) \]

```sql
select e.employee_name, city
from employee e, works w
where w.company_name = 'First Bank Corporation'
and w.employee_name = e.employee_name
```

3. Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum.

We assume each person works for at most one company. If people may work for several companies, the following solutions will only list those who earn more than $10,000 per annum from "First Bank Corporation" alone.

\[ \pi \text{employee-name, street, city} ( (\sigma (\text{company-name} = \text{"First Bank Corporation"} \land \text{salary} > 10000)) \text{works} \mid \text{employee}) \]
select * from employee
where employee name in (select employee_name
    from works
    where
        company name='First Bank Corporation'
        and salary > 10000)

4. Find the names of all employees in this database who do not work for First Bank Corporation.

The following solutions assume that all people work for exactly one company. If one allows people to appear in the database (e.g. in employee) but not appear in works, or if people may have jobs with more than one company, the problem is more complicated.

\[ \pi \text{employee-name}(\sigma \text{company-name} \neq \text{'First Bank Corporation'}(\text{works})) \]

select employee name
from works
where company name <> 'First Bank Corporation'

If people may work for several (0 or more) companies:

\[ \pi \text{employee-name}(\text{employee}) - \pi \text{employee-name}(\sigma \text{company-name} = \text{'First Bank Corporation'})(\text{works})) \]

select employee name from employee
where employee name not in (select employee name
    from works
    where company name = 'First Bank Corporation')

5. Find the names of all employees who earn more than every employee of Small Bank Corporation.

\[ \pi \text{employee-name}(\text{works}) - (\pi \text{works.employee-name} (\sigma \text{works.salary} \leq \text{works2.salary} \land \text{works2.company-name} = \text{'Small Bank Corporation'}) \rho \text{works2}(\text{works})) \]

select employee name
from works
where salary > all (select salary
    from works
    where company name = 'Small Bank Corporation')

6. Assume the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.
Note: Small Bank Corporation will be included in each answer.

company(company_name, city)

\[ \pi_{\text{company-name}}(\text{company} \div (\pi_{\text{city}}(\sigma_{\text{company-name} = \text{"Small Bank Corporation"}})(\text{company}))) \]

select S.company_name
from company S
where not exists ((select city
from company
where company_name = 'Small Bank Corporation')
except
(select city
from company T
where S.company_name = T.company_name))

7. Find the names of all employees who live in the same city and on the same street as their managers.

employee(employee_name, street, city)
manages(employee_name, manager_name)

\[ \pi_{\text{employee-name}}(\text{employee} \mid \text{manages} \mid \text{manager-name} = \text{mgr.employee-name} \land \text{employee.street} = \text{mgr.street} \land \text{employee.city} = \text{mgr.city})(\rho_{\text{mgr(employee)}}) \]

select employee_name
from employee natural join manages natural join
(select e1.employee_name, e1.street, e1.city
from employee e1, employee mgr
where manager_name = mgr.employee_name
and e1.street = mgr.street
and e1.city = mgr.city)
Exercise 2
Let the following relation schemas be given:
\[ R = (A, B, C) \]
\[ S = (D, E, F) \]
Let relations \( r(R) \) and \( s(S) \) be given.
Give an expression in SQL that is equivalent to each of the following queries.
1. \( \Pi_A (r) \)
2. \( \sigma_{B=17} (r) \)
3. \( r \times s \)
4. \( \Pi_{A,F} (\sigma_{C=D} (r \times s)) \)

Exercise 3
Consider the two following tables T1 and T2. Show the results of the following operations:

<table>
<thead>
<tr>
<th>Table T1</th>
<th>Table T2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Q</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
</tr>
</tbody>
</table>

1. \( T1 \bowtie_{T1.P=T2.A} T2 \)
2. \( T1 \bowtie_{T1.Q=T2.B} T2 \)
3. \( T1 \bowtie_{T1.P=T2.A} T2 \)
4. \( T1 \bowtie_{T1.P=T2.A \text{ and } T1.R=T2.C} T2 \)

1. 

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>A</td>
<td>5</td>
<td>10</td>
<td>B</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>5</td>
<td>10</td>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>6</td>
<td>25</td>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

2. 

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>B</td>
<td>8</td>
<td>10</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

3. 

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>A</td>
<td>5</td>
<td>10</td>
<td>B</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>5</td>
<td>10</td>
<td>C</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>B</td>
<td>8</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>6</td>
<td>25</td>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

4.