Formal Relationships Languages

Considering:
Emp (eid: integer, ename: string, age: integer, salary: real)
Works (eid: integer, did: integer, pc_time: integer)
Dept (did: integer, dname: string, budge: real, managerid: integer)

1. **Return names of every employee who works in the "Hardware", "Software", and "Research" departments. (1.5 points)**

   Data Log:

```
emp_names(ename) :- Emp(eid, ename, age, salary),
Works(eid, did1, _), Works(eid, did2, _), Works(eid, did3, _),
Dept(did1, 'Hardware', _, _), Dept(did2, 'Software', _, _), Dept(did3,
'Research', _, _).
```

   Relational Algebra:

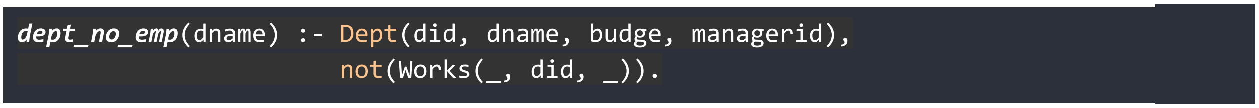
```
emp_names =  π_ename(Emp ⋈ (σ_dname='Hardware'(Dept) ⋈ Works)
                   ⋈ (σ_dname='Software'(Dept) ⋈ Works)
                   ⋈   (σ_dname='Research'(Dept) ⋈ Works))
```

   Relational Calculus:

```
emp_names = {e.ename | ∃e, w1, w2, w3, d1, d2, d3 (Emp(e) ∧ Works(w1) ∧ Works(w2)
∧ Works(w3) ∧ Dept(d1) ∧ Dept(d2) ∧ Dept(d3) ∧
e.eid = w1.eid ∧ w1.did = d1.did ∧ d1.dname = 'Hardware' ∧
e.eid = w2.eid ∧ w2.did = d2.did ∧ d2.dname = 'Software' ∧
e.eid = w3.eid ∧ w3.did = d3.did ∧ d3.dname = 'Research')}
```

2. **Return the names of every department without any employee. (1.5 points)**

   Data Log:



```
dept_no_emp(dname) :- Dept(did, dname, budge, managerid),
                      not(Works(_, did, _)).
```

   Relational Algebra:

```
dept_no_emp = π_dname(Dept) - π_dname(Dept ⋈ Works)
```

   Relational Calculus:

```
Dept_no_emp = {d.dname | Dept(d) ∧ ¬∃w(Works(w) ∧ w.did = d.did)}
```

3. **Print the managerid of managers who manage only departments with budgets greater than $1.5 million. (1.5 points)**

Data Log:

```
manager_budget(managerid) :- Dept(_, _, budge, managerid), budge > 1500000,
                             not(Dept(_, _, budge2, managerid), budge2 ≤ 1500000).
```

Relational Algebra:

```
manager_budget = π_managerid(σ_budge>1500000(Dept)) - π_managerid(σ_budge≤1500000(Dept))
```

Relational Calculus:

```
manager_buget = {d.managerid | Dept(d) ∧ d.budge > 1500000 ∧ ¬∃d2 (Dept(d2) ∧
d2.managerid = d.managerid ∧ d2.budge ≤ 1500000)}
```

4. **Print the name of employees whose salary is less than or equal to the salary of every employee. (1.5 points)**

Data Log:

```
small_salary(ename) :- Emp(eid, ename, _, salary),
                       not(Emp(_, _, _, salary2), salary2 < salary).
```

Relational Algebra:

```
small_salary =  π_ename(Emp) - π_ename(Emp ⋈_salary < salary'(ρ_salary'(Emp)))
```

Relational Calculus:

```
small_salary = {e.ename | Emp(e) ∧ ¬∃e2 (Emp(e2) ∧ e2.salary < e.salary)}
```