7.18.a. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the 'ProductX' project.

```sql
select e.Name
from EMPLOYEES e
where e.Number = 5
and exists wo in e.works_on :
    wo.Hours > 10 and wo.project.Name = 'ProductX';
```

7.18.b. List the names of all employees who have a dependent with the same first name as themselves.

```sql
select e.Name
from EMPLOYEES e
where exists dep in e.has_dependents :
    dep.name = e.Name.Fname;
```

Why is `distinct` needed only in version 2?
In version 1, each EMPLOYEES is selected only once. Thus the result is a set.
In version 2, each EMPLOYEES is paired with all DEPENDENTS. Thus a bag may result.

7.18.c. Find the names of all employees who are directly supervised by 'Franklin Wong'.

Starting with the employees.

```sql
select e.Name
from EMPLOYEES e
where e.supervisor.Name.Fname = 'Franklin'
and e.supervisor.Name.Lname = 'Wong';
```

Starting with the supervisors.

```sql
select super.supervisee.Name
from EMPLOYEES super
where super.Name.Fname = 'Franklin'
and super.Name.Lname = 'Wong';
```

7.18.d. For each project, list the project name and the total hours per week (by all employees) spent on the project.

```sql
select projName,
    sum(wo.Hours)
from WORKS_ON_PROJECT wo
group by projName, wo.project.Name;
```

Must have an instance for each employee on each project. Thus must start with WORKS_ON_PROJECT

7.18.e. Retrieve the names of all employees who work on every project.

```sql
select e.Name
from EMPLOYEES e
where for all p in (select p from PROJECTS)
    exists wo in (select wo from e.works_on :
        wo.project = p);
```

Ranges over projects for this e.
### 7.18e. Retrieve the names of all employees who work on every project.

Examine all the WORKS_ON_PROJECT for the project "p". Check employee.

```sql
select e.Name
from e in EMPLOYEES
where for all p in
  (select p from p in PROJECTS) : exists wo in
  (select wo from p.employees_on :
    wo.employee = e);
```

Ranges over projects for this p.

### 7.18e. Retrieve the names of all employees who work on every project.

Examine all the WORKS_ON_PROJECT Check employee and project.

```sql
select e.Name
from e in EMPLOYEES
where for all p in
  (select p from p in PROJECTS) :
      exists wo in
        (select wo from WORKS_ON_PROJECT :
          wo.employee = e and wo.project = p);
```

Ranges over all projects.

### 7.18f. Retrieve the names of all employees who do not work on any project.

```sql
select e.Name
from e in EMPLOYEES
where not exists wo in e.works_on;
```

### 7.18g. For each department, retrieve the department name and the average salary of all employees working in that department.

```sql
select deptName,
      avgSal : AVG(select p.e.Salary from p in partition)
from e in EMPLOYEES
group by deptName : e.works_for.Name
```

### 7.18h. Retrieve the average salary of all female employees.

```sql
avg select e.Salary
from e in EMPLOYEES
where e.Sex = 'F';
```

### 7.18i. Find the names and addresses of all employees who work on at least one project located in Houston, but whose department has no location in Houston.

```sql
select e.Name, e.Address
from e in EMPLOYEES, d in e.works_for
where exists wo in e.works_on :
    wo.project.located_at.Name = 'Houston'
and 'Houston' not in d.located_at.Name;
```

Note that d.located_at.Name is a set.
7.18.j. List the last names of all department managers who have no dependents.

```sql
select e.Name
from d in DEPARTMENTS, e in d.managed_by.manager
where not exists dep in e.has_dependents ;
```