Relational Algebra Examples

- List all pairs of Employee *names* and the project *names* they work on:

  $\Pi_{fname,lname,pname}$ (Employee $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

- List all pairs of Employee *names* and the project *numbers* they work on:

  $\Pi_{fname,lname,pno}$ (Employee $\bowtie_{ssn=essn}$ Works_On)

- List all pairs of Employee *SSNs* and the project *names* they work on:

  $\Pi_{essn,pname}$ (Works_On $\bowtie_{pno=pnumber}$ Project)

- List all pairs of Employee *SSNs* and the project *numbers* they work on:

  $\Pi_{essn,pno}$ (Works_On)

- List all projects that John Smith works on *by project name*:

  JSmith $\leftarrow \sigma_{fname=\text{“}John\text{”}\wedge lname=\text{“}Smith\text{”}}$ (Employee)
  Result $\leftarrow \Pi_{pname}$ (JSmith $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

- List all projects that John Smith works on *by project number*:

  Result $\leftarrow \Pi_{pno}$ (JSmith $\bowtie_{ssn=essn}$ Works_On)

  or

  Result $\leftarrow \Pi_{pno}$ ($\sigma_{fname=\text{“}John\text{”}\wedge lname=\text{“}Smith\text{”}}$ (Employee $\bowtie_{ssn=essn}$ Works_On))

- Find all projects (by name) that male employees work on:

  MaleEmps $\leftarrow \sigma_{sex=\text{“}male\text{”}}$ (Employee)
  Result $\leftarrow \Pi_{pname}$ (MaleEmps $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

- Find all projects (by name) that employees in department 5 work on:

  Dept5Emps $\leftarrow \sigma_{dno=5}$ (Employee)
  Result $\leftarrow \Pi_{pname}$ (Dept5Emps $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

- Find all projects (by name) that employees in “Research” work on:

  ResearchDept $\leftarrow \sigma_{dname=\text{“}Research\text{”}}$ (Department)
  ResearchEmps $\leftarrow$ (Employee $\bowtie_{dno=dnumber}$ ResearchDept)
  Result $\leftarrow \Pi_{pname}$ (ResearchEmps $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

- List all employees who work on a project that John Smith works on:

  (a) by Employee SSN

  (b) by Employee name

  (c) by Employee name (excluding John Smith himself)

  JSmith $\leftarrow \sigma_{fname=\text{“}John\text{”} \wedge lname=\text{“}Smith\text{”}}$ (Employee)

  JSProjs $\leftarrow \Pi_{pno}$ (JSmith $\bowtie_{ssn=essn}$ Works_On)

  (a) Result $\leftarrow \Pi_{essn}$ (JSProjs $\bowtie$ Works_On)

  (b) Result $\leftarrow \Pi_{fname,lname}$ (JSProjs $\bowtie$ Works_On $\bowtie_{essn=ssn}$ Employee)

  (c) Result $\leftarrow \Pi_{fname,lname}$ (JSProjs $\bowtie$ Works_On $\bowtie_{essn=ssn}$ Employee)
  $- \Pi_{fname,lname}$ (JSmith)

- List all employees who work on a project with someone who works in the same department as John Smith

  JSmithDept $\leftarrow \Pi_{dno}$ ($\sigma_{fname=\text{“}John\text{”} \wedge lname=\text{“}Smith\text{”}}$ (Employee))

  JSmithDeptEmps $\leftarrow$ (Employee $\bowtie$ JSmithDept)

  JSmithDeptEmpsProjs $\leftarrow \Pi_{pno}$ (JSmithDeptEmps $\bowtie_{essn=ssn}$ Works_On)

  Result $\leftarrow \Pi_{essn}$ (JSmithDeptEmpsProjs $\bowtie_{pno=pno}$ Works_On)

  Result2 $\leftarrow \Pi_{lname,fname}$ (Result $\bowtie_{essn=ssn}$ Employee)

- List all employee names and the department names they work for:

  Result $\leftarrow \Pi_{fname,lname,dname}$ (Employee $\bowtie_{dno=dnumber}$ Department)

- List all departments and the names of their managers:

  Result $\leftarrow \Pi_{dname,fname,lname}$ (Department $\bowtie_{mgrssn=ssn}$ Employee)

- Find all departments whose managers started managing the department at birth (their mgrstartdate equals their birthdate)

  $\Pi_{fname,lname}$ (Department $\bowtie_{mgrssn=ssn \wedge mgrstartdate=bdate}$ Employee)

  (or)

  $\Pi_{fname,lname}$ ($\sigma_{mgrstartdate=bdate}$ (Department $\bowtie_{mgrssn=ssn}$ Employee))

  (or)

  $\Pi_{fname,lname}$ ($\sigma_{mgrstartdate=bdate \wedge mgrssn=ssn}$ (Department $\times$ Employee))

- List the employees who work on **ALL** the projects that John Smith works on:

  JSmith $\leftarrow \sigma_{fname=\text{“}John\text{”} \wedge lname=\text{“}Smith\text{”}}$ (Employee)
  JSProjs $\leftarrow \Pi_{pno}$ (JSmith $\bowtie_{ssn=essn}$ Works_On)
  EmpWorksOn $\leftarrow \Pi_{essn,pno}$ (Works_On)
  ResultSSNs $\leftarrow$ (EmpWorksOn $\div$ JSProjs)
  ResultNames $\leftarrow$ (ResultSSNs $\bowtie_{essn=ssn}$ Employee)

- List the names of employees who supervise themselves

  Result $\leftarrow \Pi_{fname,lname}$ ($\sigma_{ssn=superssn}$ (Employee))

- List all the employees whose department manager earns less than $20,000.

  DeptsWithPoorMgrs $\leftarrow$
    $\Pi_{dnumber}$ (Department $\bowtie_{mgrssn=ssn}$ ($\sigma_{salary<20,000)}$ (Employee))
  Result $\leftarrow \Pi_{fname,lname}$ (Employee $\bowtie_{dno=dnumber}$ DeptsWithPoorMgrs)

- List all employees who earn more than their supervisors

  Supervisors $\leftarrow \Pi_{superssn}$ (Employee)
  SupsSals $\leftarrow \rho_{ssal=salary}$ ($\Pi_{superssn,salary}$ (Employee $\bowtie_{ssn=superssn}$ Supervisors))
  Result $\leftarrow \Pi_{fname,lname}$ ($\sigma_{salary>ssal}$ (Employee $\bowtie_{superssn=superssn}$ SupsSals))

- List all employees who earn more than their department managers

  DeptMgrs $\leftarrow \Pi_{mgrssn,dnumber}$ (Department)
  MgrSals $\leftarrow \rho_{ssal=salary}$ ($\Pi_{dnumber,salary}$ (Employee $\bowtie_{ssn=mgrssn}$ DeptMgrs))
  Result $\leftarrow \Pi_{fname,lname}$ ($\sigma_{salary>ssal}$ (Employee $\bowtie_{dno=dnumber}$ MgrSals))

- List all employees with no dependents

  AllEmps $\leftarrow \Pi_{ssn}$ (Employee)
  EmpsWithDeps $\leftarrow \rho_{ssn=essn}$ (Π_{essn} (Dependent))
  EmpsWithoutDeps $\leftarrow$ AllEmps $-$ EmpsWithDeps
  Result $\leftarrow \Pi_{fname,lname}$ (EmpsWithoutDeps $\bowtie$ Employee)

Note: In some cases above, the join condition of natural joins has been explicitly included for clarity (e.g. (Employee $\bowtie_{superssn=superssn}$ SupsSals)). This can be useful when the natural join attribute(s) are not immediately clear (e.g. (Employee $\bowtie$ SupsSals)).

Also note: In many of the above examples when creating intermediate relations, an alternate solution would include the projection of the minimal necessary attributes for subsequent use. For example, for the query "List all projects that John Smith works on":

JSmith $\leftarrow \sigma_{fname=\text{“}John\text{”} \wedge lname=\text{“}Smith\text{”}}$ (Employee)
Result $\leftarrow \Pi_{pname}$ (JSmith $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

is equivalent to:

JSmith $\leftarrow \Pi_{ssn}$ ($\sigma_{fname=\text{“}John\text{”} \wedge lname=\text{“}Smith\text{”}}$ (Employee))
Result $\leftarrow \Pi_{pname}$ (JSmith $\bowtie_{ssn=essn}$ Works_On $\bowtie_{pno=pnumber}$ Project)

The additional projections (e.g. Π_{ssn}) are not typically necessary unless subsequent joins would involve an attribute name clash.