1. What are the PIDs of the students whose name is "Suri"?
SQL:
SELECT PID
FROM Students
WHERE Name = "Suri";
Relational Algebra:
\[ \pi_{PID}(\sigma_{Name = \text{"Suri"}}(Students)) ; \]

2. Which pairs of students live at the same address? It is enough to return the names of such student pairs.
SQL:
SELECT S1.Name, S2.Name
FROM Students S1, Students S2
WHERE S1.PID < S2.PID AND S1.Address = S2.Address;
Relational Algebra:
\[ \pi_{\text{S1.Name, S2.Name}}(\sigma_{\text{S1.PID} < \text{S2.PID} \land \text{S1.Address} = \text{S2.Address}}(\rho_{\text{S1}}(\text{Students}) \times \rho_{\text{S2}}(\text{Students}))) ; \]
Here we use \( \sigma_{\text{S1.PID} < \text{S2.PID}} \) to eliminate duplicates.

3. Which departments have courses that have pre-requisites in other departments?
SQL:
SELECT DISTINCT DeptName
FROM PreReq
WHERE PreReqDeptName != DeptName;
Relational Algebra:
\[ \pi_{\text{DeptName}}(\sigma_{\text{PreReqDeptName} \neq \text{DeptName}}(\text{PreReq})) ; \]

4. Compute the set of all courses that are their own pre-requisites. The purpose of this query is to ensure that the constraint “A course cannot be a pre-requisite for itself” holds in the database. Your query needs to return only the course number and department name.
SQL:
SELECT Number, DeptName
FROM PreReq
WHERE Number = PreReqNumber AND DeptName = PreReqDeptName;
Relational Algebra:
\[ \pi_{\text{Number, DeptName}}(\sigma_{\text{PreReqNumber} = \text{Number} \land \text{PreReqDeptName} = \text{DeptName}}(\text{PreReq})) ; \]

5. What are the names and addresses of the students who are taking “CS4604”?
SQL:
SELECT Name, Address
FROM Students, Take
WHERE Number = 4604 AND PID = StudentPID AND DeptName = "CS";

Relational Algebra:
\[ \pi_{\text{Name, Address}} (\sigma_{\text{Number} = 4604 \land \text{PID} = \text{StudentPID} \land \text{DeptName} = "CS"}(\text{Students} \times \text{Take})) \];

6. What are the courses (specified by course number and department name) that the head of the CS department is teaching?

SQL:
SELECT Number, DeptName
FROM Departments, Teach
WHERE ChairPID = ProfessorPID AND Name = "CS";

Relational Algebra:
\[ \pi_{\text{Number, DeptName}} (\sigma_{\text{ChairPID} = \text{ProfessorPID} \land \text{Name} = "CS"}(\text{Departments} \times \text{Teach})) \];

7. Return the PID and names of any department head who teaches a course in another department?

SQL:
SELECT P.Name, DISTINCT PID
FROM Departments D, Professors P, Teach
WHERE ChairPID = ProfessorPID AND D.Name <> DeptName AND ChairPID = PID;

Relational Algebra:
\[ \pi_{\text{PID, P.Name}} (\sigma_{\text{ChairPID} = \text{ProfessorPID} \land \text{D.Name} \neq \text{DeptName} \land \text{ChairPID} = \text{PID}})(\rho_{\text{D}}(\text{Departments}) \times \rho_{\text{P}}(\text{Professors}) \times \text{Teach})) \];

8. Are there any students who are taking at least two courses taught by department heads? Identify these students by their PID and name.

SQL:
SELECT T1.PID, T1.Name
FROM (SELECT S.PID, S.Name, Number, DeptName
      FROM Departments, Teach NATURAL JOIN Take, Students S
      WHERE ChairPID = ProfessorPID AND StudentsPID = S.PID) T1,
      (SELECT S.PID, S.Name, Number, DeptName
       FROM Departments, Teach NATURAL JOIN Take, Students S
       WHERE ChairPID = ProfessorPID AND StudentsPID = S.PID) T2
WHERE T1.PID = T2.PID AND T1.Name = T2.Name AND (T1.Number <> T2.Number OR T1.DeptName <> T2.DeptName);

Relational Algebra:
\[ \pi_{\text{T1.PID, T1.Name}} (\sigma_{\text{T1.PID} = \text{T2.PID} \land \text{T1.Name} = \text{T2.Name} \land \text{T1.Number} \neq \text{T2.Number}}(\rho_{\text{T1}}(\text{Departments} \times \text{(Teach \bowtie Take) \times Students})) \times \rho_{\text{T2}}(\pi_{\text{T1.Name}}(\text{ProfessorPID} \times \text{StudentPID} \times \text{PID} \times \text{Number} \times \text{DeptName} \times \text{ChairPID} \times \text{(Departments} \times \text{(Teach \bowtie Take) \times Students})))) \];
9. Does the PreReq relation have cycles?
Can’t write a query for finding cycles of any length. For length 2 we can do the following:
SQL:
SELECT *
FROM PreReq P1, PreReq P2
WHERE P1.PreReqNumber = P2.number AND P1.PreReqDeptName = P2.DeptName
AND P2.PreReqNumber = P1.number AND P2.PreReqDeptName = P1.DeptName;
Relational Algebra:
\[ \sigma_{P1.PreReqNumber = P2.number \land P1.PreReqDeptName = P2.DeptName \land P2.PreReqNumber = P1.number \land P2.PreReqDeptName = P1.DeptName}(Q_1(\text{PreReq}) \times Q_0(\text{PreReq})) \]

10. A relation R has one numeric attribute A. What is the largest number in R?
SQL:
SELECT MAX(A)
FROM R;
Relational Algebra:
\[ \gamma_{\text{MAX}(A)}(R) \]

11. Which professors (specify PID, Name, and Department) earn salaries more than any department head?
SQL:
SELECT PID, Name, DepartmentName
FROM Professors
WHERE Salary > ALL (SELECT Salary
FROM Departments, Professors
WHERE ChairPID = PID);
Relational Algebra:
\[ \pi_{\text{PID, Name, DepartmentName}}(\sigma_{\text{Salary} = \text{maxsalary}}((\text{Professors} \times (\gamma_{\text{MAX}(\text{Salary}) - \text{maxsalary}}(\sigma_{\text{ChairPID} = \text{PID}}(\text{Departments} \times \text{Professors})))))) \]

12. Which professor (specify PID, Name, and Department) earns the highest salary in each department?
SQL:
SELECT PID, Name, DepartmentName
FROM (SELECT DepartmentName, MAX(Salary) as maxsalary
FROM Professors
GROUP BY DepartmentName) NATURAL JOIN Professors
WHERE Salary = maxsalary;
Relational Algebra:
\[ \pi_{\text{PID, Name, DepartmentName}}(\sigma_{\text{Salary} = \text{maxsalary}}((\gamma_{\text{DepartmentName, MAX(Salary)}} - \text{maxsalary}(\text{Professors} \bowtie \text{Professors}))) \]
13. A relation R has one numeric attribute A. The rank of a tuple t in R is the number of tuples in R whose value in A is less than the value of t in A. This question deals with computing the ranks of the tuples in R.

(a) What is the median tuple in R, i.e., if R contains n tuple, what is the tuple with rank n/2.

SQL:
SELECT R1.A
FROM R R1, R R2
WHERE R1.A < R2.A
GROUP BY R1.A
HAVING COUNT(*) = (n/2 - 1);

Relational Algebra:
\[ \pi_{R1.A}(\gamma_{R1.A \text{ COUNT} = \frac{n}{2} - 1} (\sigma_{R1.A < R2.A} (Q_{R1}(R) \times Q_{R2}(R))))); \]

(b) Compute the rank of each tuple in R.

SQL:
SELECT R1.A, (COUNT(*)+1) AS rank
FROM R R1, R R2
WHERE R1.A < R2.A
GROUP BY R1.A

Relational Algebra:
\[ \gamma_{R1.A, \text{(COUNT(*)+1) rank}} (\sigma_{R1.A < R2.A} (Q_{R1}(R) \times Q_{R2}(R))); \]

14. Assuming we have a table Numbers with a single attribute containing all the natural numbers < 100:

SQL:
SELECT A.n, B.n, C.n
FROM Numbers as A, Numbers as B, Numbers as C
WHERE C.n <= 10 AND (A.n \times A.n + B.n \times B.n = C.n \times C.n);

15. Find the name of the professor who teaches “CS4604.”
(a) Write the query in relational algebra using a natural join.
\[ \pi_{\text{Name}} (\sigma_{\text{Number} = 4604 \land \text{DeptName} = \text{"CS"}} (\text{Professors \Join_{PID = ProfessorPID} Teach})); \]

(b) Write the query in relational algebra using intersection. This version of the query has a counterpart in SQL that uses sub-queries.
\[ \pi_{\text{Name}} (\sigma_{\text{Number} = 4604} (\text{Professors \Join_{PID = ProfessorPID} Teach}) \cap \sigma_{\text{DeptName} = \text{"CS"}} (\text{Professors \Join_{PID = ProfessorPID} Teach})); \]