Handout 1 Solution

*Note: There may be more than one right answer*

1. What are the PIDs of the students whose name is "Suri"?

   SQL:
   
   ```sql
   SELECT PID
   FROM Students
   WHERE Name = "Suri";
   ```

   Relational Algebra:
   
   ```sql
   \( \pi_{PID}(\sigma_{Name = "Suri"}(Students)) \)
   ```

2. Which pairs of students live at the same address? It is enough to return the names of such student pairs.

   SQL:
   
   ```sql
   SELECT S1.Name, S2.Name
   FROM Students S1, Students S2
   WHERE S1.PID < S2.PID AND S1.Address = S2.Address;
   ```

   Relational Algebra:
   
   ```sql
   \( \pi_{S1.Name, S2.Name}(\sigma_{S1.PID < S2.PID \land S1.Address = S2.Address}(\rho_{S1}(Students) \times \rho_{S2}(Students))) \)
   ```

   Here we use \( \sigma_{S1.PID < S2.PID} \) to eliminate duplicates.

3. Which departments have courses that have pre-requisites in other departments?

   SQL:
   
   ```sql
   SELECT DISTINCT DeptName
   FROM PreReq
   WHERE PreReqDeptName <> DeptName;
   ```

   Relational Algebra:
   
   ```sql
   \( \pi_{DeptName}(\sigma_{PreReqDeptName \neq DeptName}(PreReq)) \)
   ```

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:
   
   ```sql
   SELECT Number, DeptName
   FROM PreReq
   WHERE Number = PreReqNumber AND DeptName = PreReqDeptName;
   ```

   Relational Algebra:
   
   ```sql
   \( \pi_{Number, DeptName}(\sigma_{PreReqNumber = Number \land DeptName = PreReqDeptName}(PreReq)) \)
   ```

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

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

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

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:
\[ \pi_{T1.PID, T1.Name} (\sigma_{T1.PID = T2.PID \land T1.Name = T2.Name \land (T1.Number \neq T2.Number \lor T1.DeptName \neq T2.DeptName)}(Departments \times (Teach \bowtie Take) \times Students)) \times (\pi_{T1.PID, T1.Name} (\sigma_{T1.PID = T2.PID \land T1.Name = T2.Name \land (T1.Number \neq T2.Number \lor T1.DeptName \neq T2.DeptName)}(Departments \times (Teach \bowtie Take) \times Students)))) \]

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:
\( \sigma_{P1.PreReqNumber = P2.number \land P1.PreReqDeptName = P2.DeptName \land P2.PreReqNumber = P1.number \land P2.PreReqDeptName = P1.DeptName}(\pi_{P1, P2}(PreReq) \times \pi_{P1, P2}(PreReq)) \)

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:
\( \pi_{\text{PID}, \text{Name}, \text{DepartmentName}}(\sigma_{\text{Salary} > \text{MAX}(\text{Salary})}(\text{Departments} \times \text{Professors})) ) \)

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:
\( \pi_{\text{PID}, \text{Name}, \text{DepartmentName}}(\sigma_{\text{Salary} = \text{MAX}(\text{Salary})}(\text{Departments} \times \text{Professors} >>= \text{Professors})) \);

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:
\[ \pi_{R1.A} (\gamma_{R1.A \rightarrow \text{COUNT}()} (\sigma_{R1.A < R2.A} (\rho_{R1} (R) \times \rho_{R2} (R)))) \];

(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:
\[ \gamma_{R1.A \rightarrow \text{COUNT}()} \pi_{R1.A} (\sigma_{R1.A < R2.A} (\rho_{R1} (R) \times \rho_{R2} (R))) \];

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.
\[ \pi_{\text{Name}} (\sigma_{\text{Number} = 4604 \land \text{DeptName} = \text{CS}} (\text{Professors} \bowtie_{\text{PID} = \text{ProfessorPID}} \text{Teach})) ; \]

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