HOMEWORK 3

1. **Foreign key**

   We consider the relations *Employee* and *Department.*

   $Employee(\underline{EmpId}, LastName, Dept)$

   $Department(\underline{DeptId}, Description, Location)$

   For example, the following diagram represents part of the *Employee* and the entire *Department* table.

   Employee:

EmpId	LastName	Dept
1	Slate	1
4	Flinstone	2
5	Rublle	2
7	Rockhead	3
11	Gravel	1

   Department:

DeptId	Description	Location
1	Administration	Bedrock
2	Quarry	Bedrock
3	Stockpile	Bedrock

   (a) Give a primary key for *Employee*?
   EmpId

   (b) Give a primary key for *Department*?
   DeptId

   (c) Explain why the *Dept* attribute in the *Employee* table is a foreign key referring to *DeptId* in *Department.*
   $DeptId$ is a **foreign key** of $Employee$ referring to $DeptId$ in $Department$ because:
   – *DeptId* is a candidate key of $Department$.
   – if v is a value of $DeptId$, there is a unique tuple of $Department$ in which $DeptId$ has value v.
   What does it mean for the relation *Employee* to have a foreign key?
   – One cannot create an Employee with a DeptId value that is not in Department.
   – One cannot destruct the chair of a department in Department if there are tuples corresponding to this department in Employee.

2. **Relational algebra**

(a) What is the difference between Equi-Join and Natural-Join?

Equi-join: the join condition is a conjunction of equalities.

Natural join is a special case of equi-join where the condition does not have to be explicit and the condition is on ALL attributes of the same name. In the table resulting from the natural join the duplicate columns are eliminated.

(b) Here is the relational database schema we consider.

- Student (Id: INT, Name: STRING, Address: STRING, Status: STRING)
- Professor (Id: INT, Name: STRING, DeptId: DEPTS)
- Course (DeptId: DEPTS, CrsName: STRING, CrsCode: COURSES)
- Enrolled (CrsCode: COURSES, StudId: INT, Grade: GRADES, Semester: SEMESTERS)
- Department(DeptId: DEPTS, Name: STRING)
- Teaching(ProfId:INTEGER, CrsCode:COURSES, Semester:SEMESTERS)
  NB: For example a Semester is of the form $F2000$ for Fall 2000.

(a) List the names of all the professors teaching in the history department (HIS DeptId).

$\pi_{name}(\sigma_{DeptId='HIS'}(Professor))$

(b) List the names of all the professors who taught in Fall 1994 (F1994).

$\pi_{name}(Professor \bowtie_{ProfId=ProfId} (\sigma_{semester='F1194'}(Teaching)))$

(c) List all the ProfIds of the professors who taught at least 2 courses in Fall 1999.

$\pi_{ProfId}\sigma_{tCrsCode \neq CrsCode\ AND\ Semester='F1999'\ AND\ tSemester='F1999'\ AND\ tProfId=ProfId}$
$(teaching[tCrsCode, tProfId, tSemester]X teaching)$

(d) Find the names of the courses taught in Fall 1995 together with the names of the professors who taught those courses.

$\pi_{CrsName,Name}(\sigma_{condition}(ProfessorXTeachingXCourse)$
$Condition = Teaching.ProfId = Professor.Id\ AND\ Teaching.CrsCode = Course.CrsCode\ ANDSemester =' F1995'$

(e) Find the ProfIds of the professors who taught all the courses of the CS department (DeptId CS).

Division

$\pi_{ProfId,CrsCode}(Teaching)/(\pi_{CrsCode}(\sigma_{DeptId='CS'}(Course)))$

Good work!