MODEL ANSWERS :: Examples Clinic 1

Understanding the Relational Paradigm

T1.

- **Each SECTION record is related to a COURSE record. This relationship could be called 'COMPRISES' from COURSE to SECTION, or 'BELONGS TO' from SECTION to COURSE. The identification is supported by the fact that the COURSE NUMBER attribute seems to be the primary key in COURSE and a foreign key in SECTION.**
- **Each GRADE_REPORT record is related to one STUDENT record and one SECTION record. This relationship could be called 'ACHIEVED GRADE' from STUDENT to SECTION. The identification is supported by the fact that the GRADE_REPORT seems to import the primary keys of both STUDENT and SECTION.**
- **Each PREREQUISITE record relates two COURSE records: one in the role of a course and the other in the role of a prerequisite to that course. This relationship might be called 'REQUIRES'. The identification is supported by the fact that the primary key of COURSE appears twice in PREREQUISITE.**

T2. The reason why `s_id` would not be an appropriate primary key is that it would then be possible for there to be two (or more) tuples for a single student corresponding to two (or more) advisors. The primary key should then be `s_id, i_id`.

T3.

```
I.   SELECT DISTINCT student.name
       FROM student, takes, course
      WHERE student.ID = takes.ID AND takes.course_id = course.course_id
        AND course.dept = 'Comp. Sci.'

II.  SELECT student.id, student.name
       FROM student
      MINUS
     SELECT student.id, student.name
       FROM student, takes
      WHERE student.ID = takes.ID
        AND takes.year < 2009

III. SELECT dept, MAX(salary)
       FROM instructor
   GROUP BY dept

IV.  SELECT MIN(maxsalary)
       FROM (SELECT dept, MAX(salary) maxsalary
               FROM instructor
           GROUP BY dept)
```

These are likely to be tough, particularly IV. The most important thing is to get the point that, just like in arithmetic and in relational algebra, SQL also has compositionality, i.e., an entire query can appear as a subquery inside another. Also, pay attention to typing issues: i.e., the subquery must make sense in terms of the expected type in the outer query.

There are different, equivalent ways, of answering such questions because, with SQL as with other expressive languages, one can often compute the same result with different expressions.

T4.

a. $\pi_{person\_name}(\sigma_{company\_name\ =\ 'First\ Bank\ Corporation'}(works))$

b. $\pi_{person\_name, city}(employee \bowtie (\sigma_{company\_name = 'First\ Bank\ Corporation'}(works)))$

c. $\pi_{person\_name, street, city}(\sigma_{(company\_name = 'First\ Bank\ Corporation' \land salary > 10000)}(works \bowtie employee))$

These are likely to be hard for some. In particular, there will be a tendency to get the notation wrong. For example, while there is no standardised concrete syntax for relational algebra, we try to use the Boolean symbols ∧ (for and), ∨ (for or) and ¬ (for not) but we'll not fuss too much over it.

One thing to bear in mind is how expressions are put together. Typically, it is better to work from the inside out. When there is an operation symbol (say, π), some parameter that appears as subscript (in the case of projection this is the list of columns to be projected into the result relation) and an argument relation, which, crucially, can be an entire algebraic subexpression. If you have difficulty understanding this, consider arithmetic expressions, i.e., work out for yourself that it is structurally the same as 3+(5-1), where we operate on 5-1 first in order to obtain the second argument (i.e., 4) to then perform the addition and finally yield 7. But bear in mind that some RA operators are in prefix position and some in infix position.

Another useful strategy to have is that, given a request in plain English, one should start by identifying which tables/relations are going to provide us with all the data we need. For example, while in (a) and (b), only one relation is needed, in (c) we need two. This means that a binary operator will occur in the expression, perhaps more.

*Finally, several different expressions are semantically equivalent. Just like (2*3)*5 denotes the same number as 2*(5*3), we can have different algebraic expressions that denote the same result relation.*

If here, or in T3, you came up with a different answer and want feedback on it, talk to us in the next example clinic. We'll be glad to go through it with you.