

**CIS 4301—Information and Database Systems I**  
**Fall 2007**  
**Homework 2**  
**Due Date: 09/20/07**

**Problem 1 (Exercise 3.15 in Textbook)**

Consider the Notown database: (Exercise 2.5)

Notown Records has decided to store information about musicians who perform on its albums (as well as other company data) in a database. The company has wisely chosen to hire you as a database designer (at your usual consulting fee of \$2,500/day).

- Each musician that records at Notown has an SSN, a name, an address, and a phone number. Poorly paid musicians often share the same address, and no address has more than one phone.
- Each instrument that is used in songs recorded at Notown has a name (e.g., guitar, synthesizer, flute) and a musical key (e.g., C, B-flat, E-flat).
- Each album that is recorded on the Notown label has a title, a copyright date, a format (e.g., CD or MC), and an album identifier.
- Each song recorded at Notown has a title and an author.
- Each musician may play several instruments, and a given instrument may be played by several musicians.
- Each album has a number of songs on it, but no song may appear on more than one album.
- Each song is performed by one or more musicians, and a musician may perform a number of songs.
- Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

You have decided to recommend that Notown use a relational database system to store company data. Show the SQL statements for creating relations corresponding to the entity sets and relationship sets in your design. Identify any constraints in the ER diagram that you are unable to capture in the SQL statements and briefly explain why you could not express them.

**Problem 2 (Exercise 4.3 in Textbook)**

Consider the following schema:

Suppliers (sid : integer, sname : string, address : string)

Parts (pid : integer, pname : string, color : string)

Catalog (sid : integer, pid : integer, cost : real)

The key fields are underlined, and the domain of each field is listed after the field name. Therefore *sid* is the key for Suppliers, *pid* is the key for Parts, and *sid* and *pid* together form the key for Catalog. The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in relational algebra, tuple relational calculus, and domain relational calculus:

1. Find the *names* of suppliers who supply some red part.
2. Find the *sids* of suppliers who supply some red or green part.
3. Find the *sids* of suppliers who supply some red part or are at 221 Packer Street.
4. Find the *sids* of suppliers who supply some red part and some green part.
5. Find the *sids* of suppliers who supply every part.
6. Find the *sids* of suppliers who supply every red part.
7. Find the *sids* of suppliers who supply every red or green part.
8. Find the *sids* of suppliers who supply every red part or supply every green part.
9. Find pairs of *sids* such that the supplier with the first *sid* charges more for some part than the supplier with the second *sid*.
10. Find the *pids* of parts supplied by at least two different suppliers.
11. Find the *pids* of the most expensive parts supplied by suppliers named Yosemite Sham.
12. Find the *pids* of parts supplied by every supplier at less than \$200. (If any supplier either does not supply the part or charges more than \$200 for it, the part is not selected.)

### Problem 3 (Exercise 4.5 in Textbook)

Consider the following relations containing airline flight information:

Flights(flno:integer,from:string,to:string,distance:integer, *departs*:time, *arrives*:time)

Aircraft (aid:integer, *aname*:string, *cruisingrange*:integer)

Certified (eid:integer, aid:integer)

Employees (eid:integer, *ename*:string, *salary*:integer)

Note that Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in relational algebra, tuple relational calculus, and domain relational calculus. Note that some of these queries may not be expressible in relational algebra (and therefore, also not expressible in tuple and domain relational calculus!) For such queries, informally explain why they cannot be expressed.

1. Find the *eids* of pilots certified for some Boeing aircraft.
2. Find the *names* of pilots certified for some Boeing aircraft.
3. Find the *aids* of all aircraft that can be used on non-stop flights from Bonn to Madras.

4. Identify the flights that can be piloted by every pilot whose salary is more than \$100,000.
5. Find the *names* of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.
6. Find the *eids* of employees who make the highest salary.
7. Find the *eids* of employees who make the second highest salary.
8. Find the *eids* of employees who are certified for the largest number of aircraft.
9. Find the *eids* of employees who are certified for exactly three aircraft.
10. Find the total amount paid to employees as salaries.
11. Is there a sequence of flights from Madison to Timbuktu? Each flight in the sequence is required to depart from the city that is the destination of the previous flight; the first flight must leave Madison, the last flight must reach Timbuktu, and there is no restriction on the number of intermediate flights. Your query must determine whether a sequence of flights from Madison to Timbuktu exists for *any* input Flights relation instance.