CS145 Final Examination

Tuesday, June 10, 1997, 12:15–3:15 PM

Directions

The exam is *open book*; any written materials may be used. Answer all 12 questions on the exam paper itself. The total number of points is 180, i.e., one point per minute. When doing a design in any of the models we've discussed, make the design as succinct as possible, and avoid redundancy where possible. Likewise, when asked to write queries or other database operations, you should make them as succinct as possible. All questions about SQL refer to the SQL2 or SQL3 standard, *not* to Oracle SQL.

Do not forget to **sign the pledge** below.

I acknowledge and accept the honor code.

____

Print your name here: ____

1	2	3	4
5	6	7	8
9	10	11	12

Problem 1: (30 points) We wish to design a database representing cities, counties, and states in the US. For states, we wish to record the name, population, and state capital (which is a city). For counties, we wish to record the name, the population, and the state in which it is located. For cities, we wish to record the name, the population, the state in which it is located and the county or counties in which it is located. Names of states are unique. Names of counties are only unique within a state (e.g., 26 states have Washington Counties), and cities are likewise unique only within a state (e.g., there are 24 Springfields among the 50 states). Some counties and cities have the same name, even within a state (example: San Francisco). Almost all cities are located within a single county, but some (e.g., New York City) extend over several counties.

a) In the space below, draw an entity/relationship diagram representing the above information. Be sure to indicate keys, multiplicity of relationships (e.g., many-one), and weak entity sets.

b) Convert your design from (a) into relations, using the approach outlined in the text for going from E/R to relations.

c) Represent the same information as in (a) by an ODL design. You may use abbreviations att, rel, and inv for attribute, relationship, and inverse, respectively. Do not forget to specify keys. You need not specify extents.

d) Convert your design from (c) into relations, using the approach outlined in the text for going from ODL to relations.

.

Problem 2: (15 points) The following database schema is proposed for Problem 1 (`isCapital` is a boolean, true if the city is the capital of its state):

```
Cities(cityName, cityPopulation, countyName, stateName, isCapital)
Counties(countyName, countyPopulation, stateName)
States(stateName, statePopulation)
```

a) What functional dependencies would you expect for each of the three relations?

b) What are the keys for each of the relations?

c) Are there any BCNF violations? ________ If so, would you suggest fixing them up by decomposition? If so, how? If not, why?

Problem 3: (10 points) Suppose we have a relation $R(A, B, C)$. For each of the three conditions described below, give an example of a set of functional dependencies that makes the statement true. Indicate the key(s) for R and explain briefly why the condition holds.

a) R is in BCNF.

b) R is in 3NF but not in BCNF.

c) R is not in 3NF.

Problem 4: (25 points) Suppose we have the following database schema:

```
Inventory(itemID, description, manf, quantity, price)
Manf(manfID, name, addr)
```

In `Inventory`, the five attributes are, in order:

1. A unique integer identifying the item.
2. A (not necessarily unique) description of the item, e.g., "Budweiser beer" or "short-sleeved shirt."
3. The ID of the manufacturer of the item.
4. The quantity of the item in stock.
5. The price at which the item is sold.

In the `Manf` relation, the attributes are the (unique) manufacturer's ID, name (also assumed unique), and address, respectively. Write the following queries and modifications in standard SQL2:

a) Find the names of the manufacturers of items whose description includes `beer` as a substring.

b) Find, for each manufacturer, the average quantity of their items in inventory. Manufacturers can be represented by their ID.

c) Set to 0 the quantity of all items supplied by the manufacturer `named Acme`.

d) For each value of `manf` that is found in `Inventory` but that is not a value of `manfID` in `Manf`, insert into `Manf` a tuple with that `manfID` and nulls in the other attributes.

e) Find the item ID's such that there is another item with the same description sold at a lower price.

Problem 5: (10 points) Write queries based on the database schema of Problem 4:

```
Inventory(itemID, description, manf, quantity, price)
Manf(manfID, name, addr)
```

in the language specified.

a) Find the item(s) with the largest quantity in stock [use Relational Algebra; you may use assignment (`:=`) to define temporary relations if you like].

b) Find the items mentioned in `Inventory` whose `manf` value does not appear as a `manfID` value in `Manf` [use Datalog].

Problem 6: (10 points) Write an SQL2 declaration of the `Inventory` and `Manf` relations from Problem 4:

```
Inventory(itemID, description, manf, quantity, price)
Manf(manfID, name, addr)
```

including the following constraints:

1. `itemID` and `manfID` are primary keys for their relations.
2. Manufacturer names are unique.
3. Every value of `manf` must be a `manfID` appearing in the `Manf` relation.
4. The `itemID` must be an integer in the range 0 to 32,000.
5. If the price of an item is greater than $1,000, then the quantity on hand must be less than 10.

`CREATE TABLE Manf(`

____

____

____

____

____

`);`

`CREATE TABLE Inventory(`

____

____

____

____

____

____

____

____

`);`

Problem 7: (20 points) Here is an ODL description of the inventory database from Problem 4:

```
interface Inventory
    (extent Inventories
     key itemID
{
    attribute integer itemID;
    attribute string description;
    attribute integer quantity;
    attribute float price;
    relationship Manf manf inverse Manf::items;
};

interface Manf
    (extent Manfs
     keys manfID, name)
{
    attribute integer manfID;
    attribute string name;
    attribute string addr;
    relationship Set<Inventory> items inverse Inventory::manf;
};
```

Write the following queries in OQL, using this schema:

a) Find the name of the manufacturer of the item with `itemID` 123.

b) Find the item ID's and quantities for all items manufactured by the manufacturer named Acme.

c) Find the manufacturer objects such that every item they make has at least 10 in stock (i.e., `quantity` is at least 10).

d) Find, for each item description, the total quantity in stock.

Problem 8: (10 points) A set of attributes X is said to be *closed* if $X^+ = X$; that is, no attributes except those in X are functionally dependent upon X. Suppose we have a relation $R(A, B, C, D)$ and the functional dependencies $A \to B$, $BC \to D$, and $D \to C$. Find all the closed subsets of $\{A, B, C, D\}$.

Problem 9: (15 points) Suppose that we have a graph with red and blue arcs. The graph is represented by the following two EDB relations: `Red(x,y)` means there is a red arc from node x to node y, and `Blue(x,y)` means that there is a blue arc from node x to node y. Write the following queries in recursive Datalog. In each case, let `Answer` be the predicate that is the result of your query. If you use any IDB predicates besides `Answer`, explain briefly what they mean. If you use negation, it should be stratified.

a) Find the pairs of nodes (x, y) such that there is a path from x to y that ends with a blue arc.

b) Find the pairs of nodes (x, y) such that there is a path from x to y along which colors alternate. The path must both begin and end with a red arc (possibly the same arc).

c) Find the nodes x such that there is a path from x to somewhere that ends with a blue arc, but there is no blue arc out of node x.

Problem 10: (10 points) Write the same query as Problem 9(a) using the SQL3 notation for recursive queries.

Problem 11: (10 points) Suppose we have two one-attribute relations $R(a)$ and $S(b)$.

a) Write an SQL2 assertion that says: for all x, at most one of R and S contains x.

b) It is suggested that the same effect as the assertion of part (a) could be had if we instead add a value-based check to the schema of R, as:

```
CREATE TABLE R(
    a INT CHECK(NOT EXISTS (SELECT * FROM S WHERE b=a))
);
```

Does this check have the same effect as the assertion from (a)? Why or why not?

Problem 12: (15 points) The following questions are based on two relations $R(A, B)$ and $S(B, C)$. For each, convert from the given notation to the notation specified.

a) Convert the relational algebra expression $\big(\pi_{A,C}(R \bowtie S)\big) \cup \big(\rho_{T(A,C)}(R)\big)$ to SQL.

b) Convert the Datalog rule

```
Answer(x,y) ← R(x,z) AND S(z,y) AND NOT R(x,y)
```

to relational algebra.

c) Convert the SQL query

```
SELECT MIN(A)
FROM R
```

to Datalog.