

CS 589 Principles of Database Systems

Fall 2008 Quiz 6

Monday, November 17, 2008

Suggested Answers

1. Consider the following database schema where *id* is the key for the Person table, with attributes of name and age.

Vehicle identification number (*vin*) is the key for the Car table with attributes of make (e.g., “Toyota” or “Ford”), model (e.g., “Corolla” or “Mustang”), and year.

The key for the Owns table is (*id*, *vin*) together. *Owns.id* is a foreign key that REFERENCES *Person.id*; *Owns.vin* is a foreign key that REFERENCES *Cars.vin*.

Person(id, name, age)

Car(vin, make, model, year)

Owns(id, vin)

- a. (10 points) Write a Datalog query that is equivalent to the following query.

$$\pi_{\text{name, model, year}}((\text{Person} \bowtie \text{Owns}) \bowtie (\sigma_{\text{make}=\text{“BMW”}}\text{Car}))$$

Result(n,mo,y) :- Person(i,n,a), Owns(i,v), Car(v,“BMW”,mo,y).

Most people answered this question correction. I took off one point for minor mistakes (e.g., forgetting the “BMW” part, inadvertently putting in an extra join condition, naming the variables incorrectly in some way, or writing the query in a language other than Datalog).

- b. (10 points) Write a (tuple or domain) calculus query that is NOT domain-independent.

The simplest way to write a query that is NOT domain independent is either to write an unrestricted negation or to write a disjunction (with a free variable in both disjuncts). Here’s an example of each style:

{ t | ¬Person(t) } (tuple calculus)

{ i:id, n:name, a:age | ¬Person(i, n, a) } (domain calculus)

{ n:name, v:vin | ∃i:id (∃a:age(∃ma:make(∃mo:model(∃y:year(Person(i, n, a) ∨ Car(v, ma, mo, y)))))) } (domain calculus)

In this case, if *Person(i, n, a)* is true and *Car(v, ma, mo, y)* isn’t, then *v* takes on values from the entire *vin* domain. And vice versa; if *Person(i, n, a)* isn’t true, *n* takes on values from the entire name domain.

- c. (5 points) Describe a modification to a domain (for one of the attributes in the above database) that would result in a different answer for your query in part b. (That is, describe a circumstance that would demonstrate that your query is NOT domain-independent.)

My intent with this question was for you to describe how a modification to one of the domains would result in a modified query answer. If a query answer changes when the ONLY thing you change is the definition of a domain, then we say that the query is domain-dependent. So, the important point, was for you to describe an addition (or deletion) to one of the domains in your query for part (b).

For example, if the second query that I have listed above for part (b) was my query, then if any values were added to the domain for id, name, or age, then the query answer would contain additional tuples. If the third query that I have listed above for part (b) was my query, then if any values were added to the vin domain or the name domain, then the query answer could contain additional tuples.

One style of incorrect answer was when people described how if you added additional tuples to the database (i.e., changed the database instance) then the query answer would change. (That's often true and it doesn't imply that the query was domain-dependent.)

Another style of incorrect answer was when people described how they would change the query.