

CS3431 Homework 5 (Fall 2012, B term)

Homework Due: Friday, Dec 7th, 2012.

Homework Submit: Submit in hardcopy at start of class.

Maximum Points: 50 pts

Consider a database schema consisting of three relations, whose schemas are given below. We are expected to develop your own test data sets and to perform thorough testing on these data sets. You are to test your above SQL statements also on sample data we will be providing on mypwi.

Product(model, manufacturer, type)

PC(model, speed, ram, hd, rd, price)

Laptop(model, speed, ram, hd, screen, price)

Constraints Specification in SQL. [10 pts]

Assume no primary keys nor foreign key constraints had been defined on this schema yet. Write SQL DDL to add the constraints below, whenever possible. And, demonstrate that your solution indeed works. Or if not possible, say so and explain why not.

1. Write SQL DDL to add an additional constraint to the Product table. The constraint must enforce that the type of the product must either be 'PC' or 'Laptop'.
2. Write SQL DDL to add an additional constraint that the price of any Laptop must be at least 500.
3. Add the constraint that a labtop with a larger model number must also have a higher price than one with a lower model number.
4. Add the constraint that any PC and Laptop corresponds to a model number that also exists in the Product table.
5. Add the constraint that in our product database table we only maintain products from at most 5 different manufacturers (so to assure that the quality we offer is high).

Trigger Specification in SQL. [40 pts]

For below, we assume the following relational schema with primary key constraints specified. However, the designer forgot to define any foreign keys.

```
Product(model, manufacturer, type)
PC(model, speed, ram, hd, rd, price)
FOREIGN KEY (model) REFERENCES Product(model)
Laptop(model, speed, ram, hd, screen, price)
FOREIGN KEY (model) REFERENCES Product(model)
```

Now determine how many triggers you need, what events you need to monitor, and lastly what actions are the most meaningful to take for each of the examples below. Test out your triggers, once designed, on the data set that we will provide, and how us a script illustrating the success of your tests.

1. Write one or more *triggers* to specify that for any tuple in the PC table the hard disk of the PC is at least 100 times greater than its RAM. (Note that the hard disk is in GB, while RAM is in MB).
2. Write one or more *triggers* to enforce overlap constraints, namely, to specify that when inserting a new laptop, the model number should not also appear in the PC table, and vice versa.
3. Write the needed *triggers* to enforce that whenever the prices of a Product model are being modified, that then there is a “log tuple” inserted into a special relational table (call it Product-Monitoring) that indicates the model number of the modified product listing, the type of the product (pc, printer, etc), the old price, and the new price, the time of the modification. Note that a command such as “to_char(sysdate,'dd-mm-yyyy: hh24:mi')” could be used to produce a date value. Also, remember that to first create the product-Monitoring table with the appropriate attributes.
4. Write one or more *triggers* to enforce the constraint that at all times the Product table is consistent with the other two tables. This is an extension of the foreign key constraint semantics – so now assume here that you did not have access in your DBMS to any direct support for foreign keys. That is, if in the Product table, a product row is specified as being of PC type then its model number also appears in the corresponding PC table. Similarly, if a product is of type laptop, then its model number must indeed appear in the laptop table. If the type VALUE is “NULL”, then it appears in none of the other tables. Or, vice versa, check that any tuple that is being inserted into the Laptop or the PC table, either already exists in the Product table (and, now you are simply changing its type over to the right type value), or if not to make things consistent you also have it added into the Product table as part of the current update.