**Homework 5 for CS542 - Fall 2023**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Homework Assigned: Wednesday Sept 27, 2023**

**Homework Due: Wednesday Oct 4, 2023 10 am ET**

**Submission: Submit via CANVAS. This work is to be completed INDIVIDUALLY.**

**Maximum Points: 100 pts**

**Please make sure to write your name inside the document, at the top.**

You are given a database schema consisting of three relations, whose schemas are given below. You are expected to develop your own test data sets and to perform thorough testing on these data sets using Oracle. Also test your SQL statements out on the sample data that we will provide.

**Product(model, manufacturer, type)**

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

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

**Problem 1. Constraints Specification in SQL [25 pts]**

Assume no primary keys, nor foreign key constraints have been defined on this schema yet.

Write SQL DDL statements to add the constraints below whenever possible. Do not use triggers!

Please REVIEW the supplementary Integrity Constraints Slides Attached to this homework to learn about “Check” constraints.

Demonstrate that your solution works. If you cannot enforce the constraints through DDL statements explain why.

1. Add a constraint to the Product table to enforce that the type of the product must either be ’PC’ or ’Laptop’.
2. Add an additional constraint that the price of any Laptop must be at least 500.
3. Add the constraint that a laptop 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).

**Problem 2. Trigger Specification in SQL [50 pts]**

Consider the relational schema below with primary key constraints as specified. However, the designer forgot to define any foreign keys☺

**Product(model PRIMARY KEY, manufacturer, type)**

**PC(model PRIMARY KEY, speed, ram, hd, rd, price)**

**Laptop(model PRIMARY KEY, speed, ram, hd, screen, price)**

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 provide. Turn in your script illustrating the success of your tests.

1. 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.
2. 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).
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 also appear in the laptop table. If the type VALUE is “NULL”, then it should appear 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, or if not then you also will add it into the Product table as part of the current update.

**Problem 3: View Specification in SQL [25 pts]**

Consider a database schema consisting of three relations, whose schemas are given below (the key fields are underlined). Please load data into your tables and test your answers on Oracle.

*Product( model, manufacturer, type)*

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

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

Consider the following view PCPriceList defined as:

*CREATE VIEW PCPriceList AS Select model, price from PC*

1. Use the above view to find the PC with the cheapest price. Show the SQL query.

Can you delete from this view directly? Discuss. Show how!

1. Can you perform an insert such as: INSERT INTO PCPriceList(model) VALUES (2005)? When yes and when no? Discuss.
2. What about an insert such as: INSERT INTO PCPriceList (price) VALUES (1700)? Discuss. Show what happens.
3. Now using SQL DDL, define a second view extendedPC(manufacturer, model, speed, ram, hd, rd, price, type). This view will give every PC made by each manufacturer. Can you delete from this extendedPC view? Discuss. Show what happens.