CS-A1153 Databases, Homework 4

Deadline: Wednesday, 18 May 2022 at 18:00 (late submission until Wednesday, 25 May 2022 at 18:00 with 50% of the points)

The SQL exercises will be written and submitted in A+ environment. The problem descriptions and schemas can also be found in A+. Please submit the solutions for the problems 3–5 to the designated folder in A+ as a PDF file.

Views

1. (2 p.) Consider the database of Orders. Construct a view `PopularProduct` including information of all products. The view should return the product description and total amount of units it has ever been sold. Attributes should be named as `description` and `soldInTotal`.

2. (2 p.) Consider the library database. Construct a view `ActiveLoan` including information of all active loans (the attribute `returned` is 0). The view should return the name and email address of the customer, title of the loaned item, and start and end dates of the loan. Do not rename any attributes.

Triggers

3. Consider the following database schema:

```
Property(id, address, city, year, owner, insuranceValue)
Owner(ssNo, name, address)
```

   Property ID and year are integer values. Insurance value is a float number. All the other attributes are string values.

   (a) (3 p.) Write `CREATE TABLE` commands for the relations. Add two checks: insuranceValue cannot be negative, and year must be between 1900–2100. None of the attributes can be `NULL`. Remember to define keys as well.

   (b) (1 p.) Insert at least 10 rows for the tables, so triggers can be tested in part (f).

   Write the triggers described in parts (c), (d), and (e). After that, test the triggers in part (f). Use SQLite syntax, not SQL standard syntax such that you can test the triggers using SQLiteStudio.

   (c) (1 p.) When inserting a row into table Property, check that the city is written with all caps and fix if needed.

   (d) (2 p.) When updating a row in table Property, check that the new value of insuranceValue is at least 75% of the previous value. If it is smaller, set the new value of insuranceValue to 75% of the previous value.

   (e) (2 p.) When removing a row from table Owner, check that removed person is not owner of any property in the database. If person still owns something, prevent the removal.

   (f) (1 p.) After that, write two commands for each trigger (b), (c), and (d). The other should launch the trigger and the other should not. List the contents of the tables Property and Owner both before and after running all those statements (intermediate results are not required). For example, screenshot from SQLiteStudio is enough.

Indices

4. (6 p.) Consider the relation schema `OrderContent(orderID, product, amount)` from the online store database introduced in the Exercise Round 1. Let's assume the relation occupies 120 pages of space.

   On average, each order includes 5 products and each product belongs to 30 orders. There is no clustering of any attributes.

Two kinds of queries are frequent for the table: searching for certain order ID (type Q1, fraction p1) and searching for orders with certain products (type Q2, fraction p2).

Insertions to the table take the fraction 1 - p1 - p2 of all operations on the table (type I).

Give formulas in terms of p1 and p2 to measure the cost of queries Q1 and Q2 and insertion I under the following four combinations (similarly to the exercise sessions).

- No indices at all
- Index for the attribute `orderID`
- Index for the attribute `product`
- Index for both attributes (`orderID` and `product`)

Remember to also explain where the numbers in your formulas come from.

Transactions

5. (10p.) Consider the database with the following schema

```
Product(prodID, name, price, manufID, stock)
Manufacturer(manufID, name, email)
```

In the initial state, the database contains the tuples

prodID	name	price	manufID	stock
'123'	'ABC shampoo'	3.55	'000'	50
'133'	'ABC toothbrush'	1.24	'000'	14
'422'	'Headphones'	27.99	'002'	15
'431'	'Mouse'	18.10	NULL	3

Table 1: Products

manufID	name	email
'000'	'ABC'	'customerservice@abc.fi'
'002'	'XYZ'	'xyz-services@xyz.com'

Table 2: Manufacturers

Let us inspect four transactions happening in our database. We do not know, in which order the transactions will happen, and they may occur concurrently. The commands inside a transaction will be executed in the order they are listed.

T_1(restocking):

- For all the products, read the value of the attribute 'stock'. You may assume that this happens one-by-one in an arbitrary order.
- If the value is less than 50, update the value to be increased by 20.

T_2 (ordering products):

- Read the value of 'stock' for the item with id '123'.
- If the value is at least 5, decrease the value of 'stock' by 5 (ie. the product has been ordered).
- Read the value of 'stock' for the item with id '431'.
- If the value is at least 5, decrease the value of 'stock' by 5 (ie. the product has been ordered).

T_3 (updating manufacturers):

- Update the ID of the manufacturer 'ABC' to '001'.
- Update all the items with manufID '000' to have manufID '001'. You may assume that this happens one by one in an arbitrary order.

T_4 (remove tuples breaking consistency):

- Read the results of the following query

```
SELECT prodID
FROM Product
WHERE manufID NOT IN (SELECT manufID FROM Manufacturer);
```

- Delete all the tuples in Product whose IDs are in the query.

(a) If all the ACID principles of transactions hold, which of the following scenarios are possible in the end? For possible scenarios, explain in which order the relevant transactions and their commands need to occur. If the scenario is impossible, explain briefly why.

- For the product '123' the value of 'stock' is 65.
- For the product '123' the value of 'stock' is less than 50.
- The table products contains only one tuple.
- The transaction T2 is able to order the product 'ABC shampoo' but not the product 'Mouse'.

(b) If all the ACID principles of transactions hold except atomicity, which of the following scenarios are possible in the end? For possible scenarios, explain in which order the relevant transactions and their commands need to occur. If the scenario is impossible, explain briefly why.

- The table Products contains only 1 row.
- The transaction T2 is able to order the product 'Mouse' but not the product 'ABC shampoo'.
- In the end, the Product table contains the following tuples

prodID	name	price	manufID	stock
'123'	'ABC shampoo'	3.55	'001'	45
'133'	'ABC toothbrush'	1.24	'001'	34
'422'	'Headphones'	27.99	'002'	15

(c) If all the ACID principles of transactions hold except isolation, which of the following scenarios are possible in the end? For possible scenarios, explain in which order the relevant transactions and their commands need to occur. If the scenario is impossible, explain briefly why.

- The table Products contains only 1 row.
- The transaction T2 is able to order the product 'Mouse' but not the product 'ABC shampoo'.
- The value of the attribute 'stock' for item '431' is negative.