CS-A1153 Databases, Homework 1

Deadline: April 26, 2021 at 18:00

The relational algebra statements and SQL queries 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 19–20 to the designated folder in A+ as a pdf file.

Example databases

```
LibraryCustomer (name, phone, email, libraryCardNumber)
LibraryItem (ID, loanPeriod)
Book (ID, title, author, language, year)
CD (ID, title, artist, year, length)
DVD (ID, title, year, length)
Loan (itemID, libraryCardNumber, startDate, endDate, returned).
```

The library database consists of six relations. The relation `LibraryCustomer` contains the information about library customers. Each customer is identified using the attribute `libraryCardNumber`. The relation Item contains all possible loanable items and tells the loan period for the item. Loanable items can be books, CDs, or DVDs and each have their own relation, `Book`, `CD`, and `DVD`, respectively. The `ID` in `Item` matches one `ID` in `Book`, `CD`, or `DVD`. The relation `Loan` contains the past and current information about loaned items. The attribute `returned` is 0 if the item is not yet returned and 1 if it is.

```
Customers (email, name, address, birthday)
Product (prodID, description, price, weight, type)
Orders (orderID, customer, date, payment)
OrderContent (orderID, product, amount).
```

This is the schema for the online shop. The relation `Customers` contains the information about customers, each tuple with a unique email. The relation `Product` contains the product information. The attribute `type` can be "food", "hygiene", "hobbies", or "kitchen". The relation `Orders` keeps track of the orders of the customers, with a unique `orderID`. The attribute `customer` contains the email of the customer making the order. The date of the order is in the format "YYYY-MM-DD". The attribute `payment` is "credit", "paypal", or "mobilepay". The relation `OrderContents` tells which products belong to which order, and the amount of the products in the order.

```
Doctor (id, name, email)
Patient (ssNo, name, birthday, heights, weight, phone, vaccine)
Appointment (patient, doctor, date, description).
```

These are the relations in the hospital database. The relation `Patient` contains the information about patients. The attribute `vaccine` takes the value 1 if the patient has had the corona vaccine and 0 otherwise. The relation `Doctor` contains the working ID, name, and email of a doctor. The relation `Appointment` stores the information about appointments between doctors and patients. It is assumed that a patient can meet a specific doctor only once a day. The attribute `description` in `Appointment` can take values "checkup", "acute", "treatment", and "vaccine", describing on the nature of the appointment.

Relational algebra expressions

1. (1p.) Consider the library database. Write a relational algebra expression to aswer the following query: find the title and author of all books released after the year 2000.
2. (1p.) Consider the library database. Write a relational algebra expression to aswer the following query: find the names, phone numbers, and emails of the customers that have unreturned loans.
3. (1p.) Consider the online store database. Write a relational algebra expression to aswer the following query: find descriptions, prices, and weights of the products with price lower than 5.0e and type "hygiene".

4. (1p.) Consider the online store database. Write a relational algebra expression to aswer the following query: find product IDs, product descriptions, amounts, and dates of products that appear in some of the orders.

5. (1p.) Consider the hospital database. Write a relational algebra expression to aswer the following query: find the social security numbers and phone numbers of all patients who have appointments between 26.4.2021 and 2.5.2021.

6. (1p.) Consider the hospital database. Write a relational algebra expression to aswer the following query: find the social security numbers, names, and phone numbers of patients who are born before 1961 and who have not yet been vaccinated and don't have an appointment of type "vaccine".

7. (2p.) Consider the library database. Write a relational algebra expression to aswer the following query: find the titles and return dates of items that have not yet been returned.

SQL queries

8. (1p.) Consider the library database. Write the following query in SQL: find the names and emails of all the library customers.

9. (1p.) Consider the library database. Write the following query in SQL: find the title and author of all books released after the year 2000. (Please note that this is similar to Exercise 1. How could the relational algebra expression be "translated" into SQL?)

10. (1p.) Consider the library database. Write the following query in SQL: find the name, phone and email of the owner of the library card number 842-853-8174.

11. (1p.) Consider the library database. Write the following query in SQL: find the books (all fields) that have "Harry Potter" in their title and are released after the year 2006.

12. (1p.) Consider the library database. Write the following query in SQL: find the name, phone number, and email of customers that have unreturned loans. (Please note that this is similar to Exercise 2. How could the relational algebra expression be "translated" into SQL?)

13. (1p.) Consider the library database. Write the following query in SQL: find the names of library customers who have ever loaned any book written by Leo Tolstoy.

14. (1p.) Consider the library database. Write the following query in SQL: find IDs and titles of all the loanable items with loan period over 14 days.

15. (1p.) Consider the online store database. Write the following query in SQL: find the IDs of orders containing some food item or some hobby item. List each ID only once.

16. (1p.) Consider the online store database. Write the following query in SQL: find the IDs of orders containing some food item and some hygiene item.

17. (1p.) Consider the online store database. Write the following query in SQL: find all the descriptions of the products the user with the email wiley123@yahoo.com has ever ordered, ordered alphabetically. List each item only once.

18. (2 p.) Consider the online store database. Write the following query in SQL: find the emails and names of customers who have made at least two orders with different different payment methods. List each customer only once.

Theoretical problems

19. Consider the following schema of restaurants and suppliers.

```
Restaurants(restaurantID, name, address, owner)
Suppliers(ID, name, phone, email)
Contracts(restaurantID, supplierID)
```

Each restaurant may form a contract only once with each supplier, but a restaurant can have many suppliers, and a supplier can sign a contract with multiple restaurants. Suppose that there are 100 rows in Restaurants, 50 rows in Suppliers and 100 rows in Contracts. You may suppose that there are no NULL values in the tables.

(a) (2p.) Explain briefly, what is the purpose of the following relational algebra expression, and what does it return.

$$\pi_{restaurantID,owner}(Restaurants) - \pi_{restaurantID,owner}(Restaurants \bowtie Contracts).$$

(b) (2p.) Explain briefly, what is the purpose of the following SQL query, and what does it return.

```
SELECT DISTINCT Suppliers.ID, Suppliers.name
FROM Suppliers, Contracts AS C1, Contracts AS C2
WHERE C1.supplierID = C2.supplierID AND C1.restaurantID <> C2.restaurantID AND
ID = C1.supplierID
```

(c) (2p.) What is the maximal possible number of rows in the queries (a) and (b)? What is the minimal number of rows? Please justify your answers briefly.

20. Consider the following relation and explain what is **wrong** in the proposed solutions.

```
Teacher (name, employeeNumber, email, phone)
Course (name, credits, teacher)
Student (name, studentID, email)
Grade (studentID, courseName, grade)
```

(a) (2p.) Task: List all the courses and their teachers (relational algebra).

Proposed solution: Course $\bowtie$ Teacher.

(b) (2p.) Task: Find the IDs of all students who have completed both courses Programming 1 and Programming 2.

Proposed solution:

```
SELECT studentID
FROM Grade, Course
WHERE Grade.courseName = Course.name AND (Course.name = 'Programming 1' OR
Course.name = 'Programming 2')
```