MEIC 2016/2017

Data Analysis and Integration

1st semester

Exam 1 – Solution – January 18, 2017

- The duration of this exam is **2 hours**.
- You can leave the room **1 hour** after the exam starts.
- During the exam you can access your own materials.
- Computers, tablets, or mobile phones are not allowed.
- Write your answers below the questions.
- Write your number and name at the top of each odd page.

1. (5 pts) Virtual Data Integration

1.1. (1.5 pts) Consider the following relational schema of a database named *DiseasesDB* where FK means *foreign key*:

```
Patient(patientID, lastname, firstname, birthyear, gender)
Disease(diseaseID, name)
HasDisease(patientID, diseaseID)
      patientID: FK(Patient)
      diseaseID: FK(Disease)
```

a) Write a conjunctive query that corresponds to the following SQL query:

```
SELECT p.lastname, p.firstname
FROM HasDisease hd NATURAL JOIN Disease d NATURAL JOIN Patient p
WHERE d.name = ‘cholesterolemia’
```

b) Write a conjunctive query that returns the last names and the first names of men who suffer from cholesterolemia.

c) What is the relationship between the two conjunctive queries in a) and b). Justify.

Solution:

a) Q(LN, FN) :- HasDisease(PID, DID), Patient(PID, LN, FN, BY, G), Disease(DID, ‘Cholesterolemia’)

b) Q1(LN, FN) :- Patient(PID, LN, FN, BY, ‘male’), Disease(DID, ‘Cholesterolemia’), HasDisease(PID, DID)

c) Q contains Q1, because it Q1 only returns the last names and first names of men who suffer from cholesterolomia.

1.2. (1.5 pt) Consider the following mediator schema of a data integration system that uses the database DiseasesDB as data source:

Diabetes(PID) $\supseteq$ DiseasesDB.HasDisease(PID, DID), DiseasesDB.Disease(DID, ‘Diabetes’)
MalePatientsUnder40(PID, LN, BY) $\supseteq$ DiseasesDB.Patients(PID, LN, FN, BY, ‘male’), BY > 1977

a) What type of schema mapping (Global-As-View or Local-As-View) does it represent? Justify.

b) How would you express the following query using the mediator schema: Which patients (patient id and last name) are male, 30 years old, and suffer from diabetes?

c) Could you answer a variation of b) where the age of patients returned is 45 using the mediator schema? Justify.

Solution:

a) GAV because the tuples of relations Diabetes and MalePatientsUnder40 are defined as supersets of query expressions (or views) over the database relations.

b) Q2(PID, LN) :- Diabetes(PID), MalePatientsUnder40(PID, LN, BY), BY = 1987

c) Q3(PID, LN).
No, we cannot because the mediator schema is only able to return patients whose age is less than or equal to 40, so it does not return patients aged 45.

1.3. (1 pt) Suppose you have the following pre-computed view defined over the tables of DiseaseDB:

PatientsWithDiabetes (PID) :- HasDisease(PID, DID), Disease(DID, 'diabetes')
FemalePatients(PID) :- Patient(PID, LN, FN, BY, 'female')

Could you use them exclusively to answer the query: return the last names of the females who suffer from diabetes and are aged under 40?

Solution:

No, we cannot, because the schema of the views PatientsWithDiabetes and FemalePatients does not include the birth year, so we do not have a way of imposing the condition that the patients females have to be aged under 40. Moreover, none of the views returns the last names of patients as it is required in the query.

1.4. (1 pt) Give an example of an application for which the virtual data integration architecture makes sense and another example for which a data warehouse architecture makes sense. Recall the advantages and inconvenients of both types of a data integration architecture.

Solution:

Virtual data integration: when the integrated data needs to be as fresh as possible => query over online data (for example, web sites publishing data about airline tickets)
Data warehouse: when applications need to compute aggregations (or data mining algorithms) over source data that has been integrated (and store historical data) and it is not relevant to obtain exact data. One possible application domain is data about customers, products and sales over time in a company and the goal is to compute aggregations over the sales.

2. (5 pts) Data Cleaning

Consider the database DiseaseDB of the previous exercise and the Patient table. At a given point in time, users of this database notice that it contains several records that refer to the same patient even though the values of the attributes ID, firstname, lastname, etc can be slightly different. Therefore, a data cleaning process is required.

2.1. (1.5 pts) In order to find out the Patient records containing similar names, the IT department of the hospital wants to try several string matching algorithms. One of them is the Jaro measure, because this measure is known as being effective to determine similar person names.

Compute the Jaro measure between the two surnames:

Elmagramid
Emalgamid

Solution:

X = Elmagramid
Y = Emalgamid

|X| = 10
|Y| = 9
min(|X|, |Y|)/2 = 9/2 = 4.5

Common characters: elmagamid (9)
Transposed characters: (3)
Elmagamid
Emalgamid

Jaro (X, Y) = 1/3 [c/|x| + c/|y| + (c –t/2)/c] = 1/3 * (9/10+9/9 +(9-3/2)/9) = 0.91

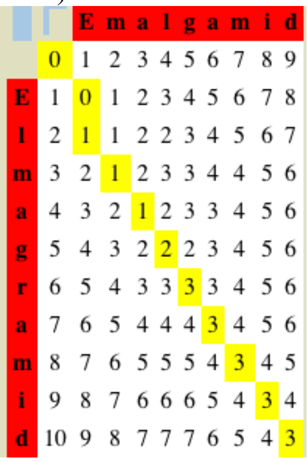
2.2. (1.5 pt) The other string matching algorithm that the Hospital IT department wants to try is the edit distance and then they want to compare the results obtained with the ones obtained with the Jaro measure.

a) Do you think the results of the two measures, edit distance and Jaro measure, are comparable? Justify. If the answer is no, what can be done in order to compare them?
b) Compute the edit distance between the two strings above.

Solution:

a) No, they are not comparable, because edit distance is an integer and the Jaro measure is a number between 0 and 1. We can convert the edit distance value into a similarity and then they are comparable.

b) Edit distance = 3



		E	m	a	l	g	a	m	i	d
	0	1	2	3	4	5	6	7	8	9
E	1	0	1	2	3	4	5	6	7	8
l	2	1	1	2	2	3	4	5	6	7
m	3	2	1	2	3	3	4	4	5	6
a	4	3	2	1	2	3	3	4	5	6
g	5	4	3	2	2	2	3	4	5	6
r	6	5	4	3	3	3	3	4	5	6
a	7	6	5	4	4	4	3	4	5	6
m	8	7	6	5	5	5	4	3	4	5
i	9	8	7	6	6	6	5	4	3	4
d	10	9	8	7	7	7	6	5	4	3

2.3. (1 pt) After having applied an approximate duplicate detection operation to the Patient table, the result is the following table: `PatientMatches(ID1, lastname1, firstname1, birthyear, gender1, ID2, lastname2, firstname2, birthyear2, gender2)`. This table stores pairs of Patient records that were considered as being similar enough according to some criteria.

The goal now is to bring together all Patient records that are potential duplicates. Can you think of an algorithm that accepts as input the `PatientMatches` table that is able to do that? Write the corresponding pseudo-code or briefly explain how it works.

Solution:

The algorithm that could be applied to the relation PatientMatches is Transitive Closure.
It will return a relation with two attributes: cluster Id and Patient ID.
The goal is to put in the same cluster (i.e., assign the same cluster ID value) to all tuples that potentially refer the same real world entity.

This algorithm would analyse each tuple of the PatientMatches table and would associate the same cluster ID value to the two IDs that are related in a tuple of PatientMatches.
If a Patient ID' is already associated to a clusterId, it will assign the same clusterID to the Patient ID'' that is paired with Patient ID'

2.4. (1 pt) Since all attributes of the Patient table may have data quality problems, it is not enough to compare their last names. Suppose you use the Jaro measure to compare the last/first name attributes. Furthermore, assume that two Patient records are considered similar only and only if their birth dates are equal and their gender values are equal.

a) Suggest the pseudo-code of a rule to decide whether two Patient records r1 and r2 are similar taking into account this information.
b) State an advantage and an inconvenient of this method for comparing Patient records.

Solution:

a) For example, the rule:

```
    if Jaro(r1.lastname, r2.lastname) >= 0.9
    and Jaro(r1.firstname, r2.lastname) >= 0.8
    and r1.birthyear = r2.birthyear
    and r1.gender = r2.gender
    then r1 is similar to r2
```

b) Advantage: we can specify in which conditions a pair of records is similar, it is easy to understand
Inconvenient: it is labor intensive to write good rules.

3. (5 pts) Data Warehouse and ETL process

A data warehouse stores historical information about the cholesterol measurements on patients who are taking a certain medicine (drug). The multidimensional model has one fact table and three dimensions:

• D_Patient	with descriptive attributes	patient_name, birth_year, gender;
• D_Time	with descriptive attributes	date, month, week, year;
• D_Medicine	with descriptive attributes	medicine_name, medicine_class, dosage.

The D_Time dimension contains the hierarchy date -> week -> month -> year, and the D_Medicine dimension contains the hierarchy: medicine_name -> medicine_class. Each tuple in the fact table, named F_Chol_Parameters stores the values of HDL Cholesterol (High-density Lipoprotein Cholesterol) and LDL (Low-Density Lipoprotein Cholesterol) measured for a certain patient taking a certain medicine, in a given instant of time.

3.1. (2,0 pts)

a) Present the star relational schema of this data warehouse using the notation:

Relation1(primary-key, att1, att2,…)
att2: FK(Relation2)

where *FK* means *foreign key*. Consider surrogate keys for the dimension tables.

b) Which aggregation function would make sense to use for aggregating HDL and LDL values through the dimensions D_Medicine and D_Time? Do you think that *sum()* would make sense? Justify your answer.

c) Taking into account the answer given to the previous question, classify the measures HDL and LDL as additive, semi-additive or non-additive. Justify your answer.

d) Since one week may belong to two different months, what can you say regarding the D_Time dimension and the fact that you may want to aggregate values of HDL and LDL through time? (Hint: Recall the notion of summarizability)

Solution:

a) D_Patient(Patient_ID, patient_name, birth_year, gender)
D_Time(Time_ID, date, month, week, year)
D_Medicine(Medicine_ID, medicine_name, medicine_class, dosage)
F_Chol_Parameters(Patient_ID, Time_ID, Medicine_ID, HDL, LDL)
Patient_ID: FK(D_Patient)
Time_ID: FK(D_Time)
Medicine_ID: FK(D_Medicine)

c) Average() would make sense as an aggregation function. Sum() would not make sense, since we do not add the HDL/LDL values through time, not Patient, nor Medicine.

d) The measures are non-additive, because they cannot be summed up through any of the dimensions.

e) We cannot aggregate the HDL/LDL values through the time dimension from week to month. A week may belong to two distinct months, so if we computed the average of the HDL/LDL values, we would count twice the value of HDL/LDL in that week for two different months which is wrong.

3.2. (1,5 pt) Consider the following SQL query:

```
SELECT avg(HDL), avg(LDL)
FROM F_Cholesterol f NATURAL JOIN D_Medicine m NATURAL JOIN D_Patient p
GROUP BY m.medicine-class, p.patient-name
```

If you replace `GROUP BY m.medicine-class, p.patient-name`, with:

a) `GROUP BY ROLL UP m.medicine-class, p.patient-name`, what would be the difference in the results obtained?
b) `GROUP BY CUBE m.medicine-class, p.patient-name`, what would be the difference in the results obtained?
c) `GROUP BY GROUPING SETS ((m.medicine-class), (p.patient-name))`, what would be the difference in the results obtained?

Solution:

a) The query would return the union of the results of SQL queries with: GROUP BY medicine-class, patient-name, GROUP BY medicine-class, and no GROUP BY.
b) The query would return the union of the results of SQL queries with: GROUP BY medicine-class, patient-name, GROUP BY medicine-class, GROUP BY patient-name, and no GROUP BY.
c) The query would return the union of the results of SQL queries with: GROUP BY medicine-class, GROUP BY patient-name

3.3. (0.5 pt) What would be the difference in terms of the results obtained with respect to the SQL query given in Question 3.2, if there was no GROUP BY clause?

Solution:

The query would return the total average values of HDL and LDL

3.4. (1 pt) Suppose the data for populating the D_Patient dimension comes from the database DiseasesDB Patient table. This table contains dirty data, in particular approximate duplicate records. Before loading the D_Patient dimension, you want to run a procedure that detects the approximate duplicate records.

State which of the following statements are true. For those statements that you mark as false, provide a justification for your answer.

a) The use of a surrogate key in the D_Patient dimension eliminates the approximate duplicate records.
b) To detect approximate duplicate records in the Patient table, we need to compare every pair of records, but we do not need to compare a record with itself.
c) To detect approximate duplicate records in the Patient table, it is always true that we only need to compare the name and the gender attributes.
d) The procedure to detect approximate duplicate records in the Patient table cannot be programmed in SQL.

Solution:

a) False. A distinct surrogate key is assigned to each dimension record, so if there are two records that concern the same real world entity, they will be assigned two distinct surrogate key values.
b) True.
a) False. The attribute Birth_year may also contain errors.
b) False. It can be programmed in SQL if the DBMS enables to invoke external functions within SQL instructions.

4. (5 pts) Data Integration Tools

Consider the following database schema of medical appointments:

```
Patient(patientID, name, city, country)
Physician(physicianID, name, specialty)
Appointment(patientID, physicianID, datetime)
      patientID: FK(Patient)
      physicianID: FK(Physician)
```

4.1. (1,5 pt) In this database, some countries are spelled in multiple different ways (e.g. U.S., USA, United States, etc.). We want to replace those different occurrences with a single designation. There is a CSV file with the adopted designation for each country:

possible designation	adopted designation
U.S.	United States
USA	United States
United States	United States
…	…

Draw a PDI transformation to replace each country with the adopted designation given in the CSV file. Indicate all the steps that you would use, and briefly describe their purpose.

Solution:

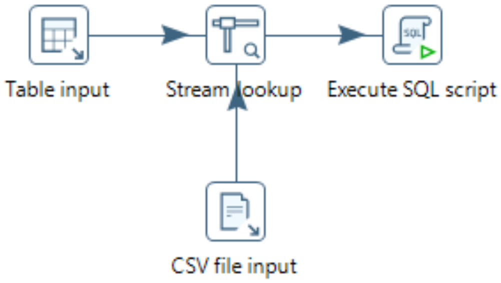


Table input to read the Patient table.
CSV file input to read the CSV file.
Stream lookup to get the adopted designation for the country coming from the Patient table.
Execute SQL script to perform an UPDATE on table Patient in order to replace the country designation.

4.2. (1 pt) How could DataCleaner help you with the same problem? What would be able to do with DataCleaner with respect to this problem of country names? Explain in your own words.

Solution:

DataCleaner is actually a data profiling tool. It could help us figure out which values are present in the country column, possibly get some statistics for those values (shortest, longest, NULLs, etc.). However, to actually replace the data, we would need to use a tool such as PDI.

4.3. (1,5 pt) From the database above, someone created a data warehouse with a fact table called `fact_appointments` and dimension tables `dim_patient`, `dim_physician`, and `dim_time`. The goal is to

analyze the number of appointments according to these dimensions.
Suppose you are using PSW (Pentaho Schema Workbench) to define the data cube. Draw the entire tree structure of the cube definition as it would appear in PSW.

Solution:

- (Cube) Appointments
    - Table: fact_appointments
    - (Dimension) Patient
        - Patient Hierarchy
            - Country
            - City
            - Name
            - Table: dim_patient
    - (Dimension) Physician
        - Physician Hierarchy
            - Specialty
            - Name
            - Table: dim_physician
    - (Dimension) Time
        - Time Hierarchy
            - Year
            - Month
            - Day
            - Table: dim_time
    - (Measure) number_of_appointments

4.4. (1 pt) Suppose you are using Saiku Analytics to query the data warehouse. How do you perform a query in Saiku Analytics? What is the relationship between using Saiku Analytics and querying the data warehouse with MDX? Explain in your own words.

Solution:

In Saiku, the query is performed by dragging a measure to Measures, and dragging dimensions (i.e. hierarchy levels) to either Rows or Columns. Behind the scenes, Saiku generates the code for an MDX query, and executes it over the data warehouse. So Saiku is actually a front-end for executing MDX queries.