Data Analysis and Integration

Exam – February 2, 2018

1. Suppose that you have two data sources A and B, and you want to define a mediated schema.
   a) How do you decide which attributes of A and B should be part of the mediated schema, and which attributes should not?

      In the mediated schema, we should have the common attributes of A and B. Any specific attributes that exist only in A or B (but not in both), should not be part of the mediated schema.

   b) If you have separate queries over each data source, how do you combine these queries in order to answer a query over the mediated schema? Give an example.

      The queries can be combined into a view with the SQL union operator. For example, if both A and B contain employees, we could define: create view all_employees(id, name) as (select emp_no, emp_name from A.employees) union (select emp_id, emp_name from B.employees).

   c) Suppose you have two queries over the mediated schema. How can you prove (or disprove) that they are equivalent?

      First, we have to unfold both queries. Then we should check if there are redundant subgoals, which can be removed. Finally, if both queries are satisfiable and have the same subgoals, then they are equivalent.

2. Someone said that everything that can be done in an ETL tool (such as PDI) can also be done in SQL.
   a) How would you convince that person that it is better to use an ETL tool? Present two advantages.

      PDI offers a large set of operations that we can pick and use right away in a transformation without having to implement them in SQL. In addition, a PDI transformation is a more explicit way to define and maintain ETL processes. In SQL, it could be difficult to implement these transformations, since they would lead to very large and complex queries.

   b) Give an example of an operation that can be done in PDI but would be much faster in SQL. In the labs, what was the purpose of using such operation?

      Joining records from multiple tables is an example of an operation that can be done much faster inside the database system with SQL. In the labs, we used a Join Rows step to compare records in order to detect approximate duplicates. We had to do the join operation in PDI because the input data was in a CSV file.

   c) What is the difference between a PDI transformation and a PDI job? Explain the concept of PDI job.

      A PDI transformation is a pipeline of processing steps that work in streaming mode. In contrast, a PDI job is a sequence of PDI transformations, where each transformation is executed only after the previous transformations have successfully completed.

3. In a data profiling process, it is possible to use DataCleaner and PDI together.
   a) What kind of operations would you do in PDI, and what kind of operations would you do in DataCleaner? Give an example of an operation in each tool.

      PDI is especially useful to extract and transform data in a way that we can analyze. On the other hand, DataCleaner is a tool that can be used to detect anomalies and gather statistics about the data. We could apply these statistics to the output of a PDI transformation. For example, we could use a Calculator step in PDI, and then analyze the output with a Value Distribution in DataCleaner.

4. Suppose you develop a transformation to detect approximate duplicates in a customer database (with first name, last name, e-mail, phone).

a) What would you change in the transformation if someone told you that the e-mail and phone are more reliable to identify a customer? Be specific about the step(s) that you would change and how.

*The total similarity between two records is calculated based on a weighted sum, such as: sim_total = 0.2*sim1 + 0.3*sim2 + 0.2*sim3 + 0.3*sim4. If the e-mail and phone are more important, then we could give more weight to their similarity and less weight to the similarity of other attributes. This could be done by changing the Formula step in PDI.*

b) Suppose you use Jaro-Winkler. What is the danger of setting a too high threshold or a too low threshold in the transformation? Explain.

Jaro-Winkler is a similarity measure. If the threshold is too high, we will be very restrictive and probably we will reject pairs which contain true duplicates (these would be false negatives). On the other hand, if the threshold is too low, we will possibly accept pairs which are not truly duplicates (these would be false positives).

c) You find that customers A and B are duplicates, as well as B and C. However, A and C do not match. Should A and C be considered duplicates or not? Justify.

According to the principle of transitive closure, if A and B and duplicates, and B and C are duplicates, then A and C should be considered duplicates as well. This would mean that A, B and C would form a cluster of duplicates.

Answer the following questions in a separate sheet of paper

5. Suppose you are designing a data warehouse.
   a) If you use a star schema, how do you decide how many dimension tables there will be? Give an example to illustrate your answer.

   The number of dimension tables is decided based on the dimensionality of facts that are stored in the data warehouse. For example, if the facts are in the form "customer X bought product Y on date Z", then we need three dimensions: customer, product and time. In a star schema, there is a single table for each dimension, so this would lead to three dimension tables.

   b) Is it possible to query a data warehouse with SQL? How can you perform operations such as roll-up and drill-down with SQL?

   A data warehouse can be stored in a regular database system such as MySQL. Therefore, it is certainly possible to query it with SQL. To perform roll-up or drill-down operations, we should use an appropriate GROUP BY. For example, if we do "GROUP BY country" and then change it to "GROUP BY country, city" we are performing a drill-down operation. If we do the opposite change, we are performing a roll-up.

   c) Give an example of a parallel (or alternative) hierarchy. How would you implement such hierarchy in the data warehouse schema?

   A parallel or alternative hierarchy contains more than one possible roll-up path. For example, we can aggregate months either by calendar year (Jan-Dec) or by academic year (Sep-Aug). Such hierarchy is implemented with a snowflake structure, where the Month table has two foreign keys, one to the CalendarYear table, and another to the AcademicYear table.

6. In this course, you were introduced to the concept of slowly-changing dimensions (SCDs).
   a) How does adding validity intervals (such as from_date, to_date) help when dealing with SCDs?

   These validity intervals allow us to store multiple versions of the same record. Each version is valid from a start date (from_date) to an end date (to_date). The current version is the one for which the current date is

between from_date and to_date. For any given version, its from_date is equal to the to_date of the previous version; it is the date when a change occurred and therefore the previous version expired.

b) Instead of using validity intervals, another approach is to have two columns for each attribute that may change. What is the disadvantage of this approach?

With two columns, we can only store the current version and the previous version. So we can have only two versions for each record, and without timestamps. So the disadvantage is that we lose track of older versions and also of the time when the changes occurred.

c) Would you say PDI has good support for SCDs? Why? Justify your answer.

PDI supports SCDs through the Dimension Lookup-Update step. This step is used in the transformation that populates the SCD dimension table. It automatically creates the values for the surrogate key, for the version number, and for the from_date and to_date fields. So PDI has good support for Type 2 SCDs.

7. To define the OLAP cube, we typically use a tool such as Pentaho Schema Workbench (PSW).
   a) In the time dimension, we usually have two attributes for month (month_id and month_name). What do they contain and why do we need them both?

   Month_id contains the month number from 1 to 12. Month_name contains the corresponding month name from January to December. In the cube definition, we need to use both, because month_id is used for sorting purposes and month_name is used for display purposes. This can be seen in the XML cube definition where we can find the attributes column=" month_name" and ordinalColumn=" month_id".

8. To analyze the data in a data warehouse, we use can use MDX, reports and KPIs.
   a) What is the MDX code equivalent to dragging two dimension levels to the COLUMNS in Saiku?

   If the levels belong to the same dimension:
   SELECT {Customer.Country.Members, Customer.City.Members} ON COLUMNS
   If the levels belong to different dimensions:
   *SELECT Customer.Country.Members * Time.Year.Members ON COLUMNS*

   b) What is the purpose of the WITH clause in MDX? Explain and give an example.

   The WITH clause can be used to define calculated measures or named sets to be used in a query. These are defined at the top of the query in order to simplify the SELECT expression. An example is:
   WITH MEMBER Measures.SalesPerUnit AS (Measures.Sales / Measures.Quantity)
   SELECT Measures.SalesPerUnit ...

   c) Suppose you have calculated a KPI with an MDX query. How do you determine if the KPI is indicating good or bad performance?

   A KPI has a target value that is defined by business managers, and has a current value that can be calculated from data. To determine if the KPI is indicating good or bad performance, we need to compare its current value to the target value. When a target value is not explicitly defined, we can compare the current value with the value calculated for a parallel period (previous year or previous month, for example).