Databases and Data Mining 2017
Final Exam

Gorlaeus Room G03
Friday December 22nd 2017
10.00 – 13.00

- State your name and student number and affiliation (e.g. LU (Leiden University), TUD (TU Delft), etc.) on every page of your answers.
- Every assignment has the same weight. There are 10 assignments.
- **Always fully motivate and explain your answers!** (Answering a question with only 'yes' or 'no' will never be counted as a correct answer.) Use examples and/or sketches whenever / wherever you consider them useful to explain your answers.
- Please note that you have a total of 3 hours to answer the questions.
- It is an open book exam: you are allowed to use your book and course notes (slides).
- **All electronic equipment should be off the table and switched off.**

1. "Classical" data warehouse techniques focus on efficiently precomputing and materializing (crucial parts of) data cubes, trading in higher memory/storage requirements to reduce (repetitive) computation costs. Discuss why this is more important in traditional row-store relational database systems, while modern column-store database systems, like, e.g., MonetDB, often can "afford" not to precompute and materialize any data cubes.

2. Name and explain three typical ways of modeling data in data warehouses. Categorize the tables involved as well as the attributes per table category and discuss the role of each table category and each attribute category.

3. Consider the following relational table schema representing a four-dimensional base cuboid with one measure per cell:

```
CREATE TABLE t (
string  d1, -- dimension 1
integer d2, -- dimension 2
string  d3, -- dimension 3
integer d4, -- dimension 4
real    m   -- measure
);
```

   Give a set of SQL queries that calculate all cuboids of the entire data cube, using `sum()` as aggregation function.

4. Complete data cubes are rather large, in particular with high-dimensional date, and thus expensive to compute and store. Name and explain two techniques to calculate and store only partial data cubes and discuss their intention, effectiveness (in terms of reducing the computation and storage requirements) and limitations.

5. Explain a (one) technique of your choice to efficiently calculate data cubes. Discuss strengths and weaknesses/limitations of this technique.

6. Assume a supermarket database *DB* contains all customer transactions. Each transaction consists of the set of items bought by the customer at a single visit. The items bought are from a very large item set {A, B, C, …}. (Note: *A* can be something like *"milk", B = "bread", C = "sugar", etc*.) After data mining the database the pattern *A => B* is found.

   a) Give two examples of null-invariant interestingness measures for the correlation/dependence of the two items/events *A* and *B*.
   b) What values of your measures would you expect, if the items/events *A* and *B* are positively correlated/dependent in *DB*?
   c) What values of your measures would you expect, if the item sets in *DB* are imbalanced and positive with respect to items/events A and B?

7. A database *DB7* has five transactions. Let min_sup = 50%.

TID	*Items_bought*
T100	{A,C,E}
T200	{D,E,F}
T300	{A,B,C,D,F,G}
T400	{A,B,C,D,E,G,H}
T500	{A,C,D,E,H}

   a) Find all frequent item-sets using the Apriori-Algorithm (give the intermediate results for the different steps of the algorithm).
   b) If the items in a transaction are taken from a set of 1000 different items, what would be the maximum number of database scans the Apriori Algorithm has to execute in order to calculate all possible frequent item sets? Which method improves on this?

8. Consider the database *DB7* given in Assignment 7.
   a) Assume we determine the frequent items using the FP-Growth algorithm and min_sup = 30%. Give the *f-list* for the database *DB7*.
   b) Assume that the prices of the items A, B, C, D, E, F, G, and H are 1, 2, 8, 3, 15, 6, 100, and 4 euro, respectively. Give an example of a strongly convertible constraint on the prices of the items.
   c) Describe how your constraint can be exploited in frequent item set mining when using the FP-Growth algorithm.

9. An optimization in frequent item set mining is mining max patterns.

   a) Describe why mining max patterns can be done more efficiently than mining frequent item sets.
   b) Would the FP-Growth method be useful for mining max patterns in a 10000-dimensional data set of 10^7 elements?
   c) What are 2 main differences between max and closed patterns?

10. *LinkedIn* is a business- and employment-oriented social networking service that is used for professional networking, Let a graph *G = (V, E)* be defined by *V* the set of vertices being equal to the set of *LinkedIn*-members, and *E* the set of edges between vertices of *V*, where an edge *(v, w)* is in *E*, if and only if the *LinkedIn*-member represented by vertex *v* has a professional relationship with the *LinkedIn*-member represented by vertex *w*.
    a) Give 3 important graph-theoretic characteristics of graph *G*. Explain why they are important and why they are characteristic for this kind of graphs.
    b) Give 2 other examples of natural occurring networks that also have these characteristics.
    c) Describe an algorithm that can be used to produce a synthetic graph *H* with at least 2 similar graph-theoretic characteristics as *G* (out of the three you mentioned).