INFOH419 Data Warehousing January 2016
prof. Toon Calders

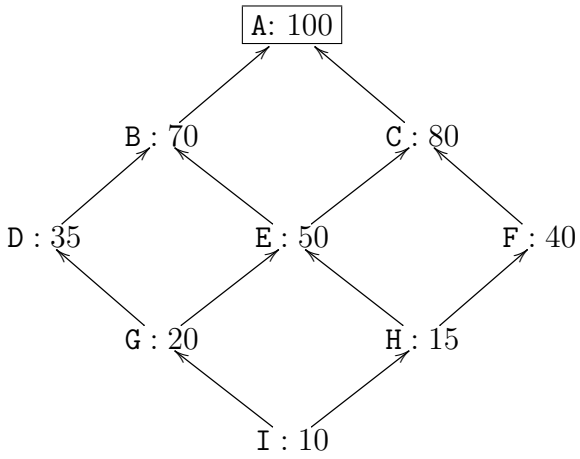
The exam is **open book**, so all books and notes can be used. "Open book" implies that you can refer to a specific slide, book page, or exercise. Verbatim copying lecturing material will not be rewarded. Stay focused on the question and avoid excessively long answers; succinct, to the point answers will be rewarded.

The questions in the exam should all be self-explanatory. In case of doubt, make a short note and solve the question to the best of your knowledge.

The maximal time to complete this exam is 3h and will be strictly observed. Plan your exam accordingly.

---

1. (5p) An important technique to speed up analytical queries is by pre-computing and materializing aggregations. Consider the following lattice of views that can be requested by the user, along with the size of each view.



   A is the view representing the base relation. The edges indicate the relation "can be computed from."

   (a) Suppose that only the top-view `A` has been materialized. Select 4 additional views from the views `B`, `C`, `D`, `E`, `F`, `G`, `H`, and `I` to materialize. Apply the greedy method described by *Harinarayan, Rajaraman, and Ullman* in their seminal paper "Implementing Data Cubes Efficiently" (SIGMOD 1996) in order to optimize the overall query time.

   (b) What is the gain under the cost model of *Harinarayan et al.* that you obtain by materializing these 4 views?

2. (3p) Explain the benefits of the bitmap-join index for data warehouses and give an example of a query that would benefit from such an index (give the relevant part of a star-schema, on which table(s) and attribute(s) the index is built, the query and an explanation of the gain obtained by the index). Then explain why such indices are less suitable for OLTP databases.

3. (6p) Consider the following description of a data warehouse. Create a dimensional fact model that captures the main characteristics of the description. In case of doubt, please follow your intuition; the essential points in the description have been clearly stated.

   The homeland security department of a country wants to organize all information of its visitors in one central data warehouse. In this data warehouse the following information will be stored for all visitors:

   - *Personal details: name, date of birth, city and country of residence and of birth, family situation (single/married with or without children);*
   - *The start and end date of the visit, # days the visit lasted;*
   - *Approximate worth of the luggage when entering and leaving the country;*
   - *The Visa that was used (if any—not all nationalities need a visa): number of the visa, start and end date of the visa, visa type.*

   All this information needs to be stored in the data warehouse both for ongoing and finished visits. Furthermore, the data warehouse should allow queries such as: "give the total number of visitors per month for the period 2010 till 2015", "Give the number of visitors with a type B visa that started their visit in January or February (any year)", "Give the total number of visit days per country of residence and visa type over the past 5 years."

4. (2p) *Jensen et al.* make in their book *Multidimensional Databases and Data Warehousing* the following statement regarding the use of a surrogate key as the primary key in a dimension table of a star-schema:

   > [A surrogate key as key column in a dimensional table] has several advantages over the option of information-bearing keys from the source systems, including better storage use, prevention of problems associated with key re-use, better support for dimension updates, and more efficient query processing.

   Explain, **in your own words, and with an example of your own** the second and the third benefit; i.e., (a) why a surrogate key prevents problems associated with key re-use, and (b) why it better supports dimension updates.

5. (4p) Consider the following (partial) star schema of a data warehouse storing sales information. The primary keys in the relations have been underlined.

   - FactSalesLine(CID, PID, DID, OrderID, quantity, unitPrice, discount, totalPrice)
   - dimClient(CID, name, dob, address, segment, VIP, creditscore, start, end)
   - dimProduct(PID, ...)
   - dimDate(DID, ...)

   As can be seen, the dimClient dimension is versioned (start and end attributes are added to it) in order to capture type-2 changes for the address, segment, VIP status, and creditscore attributes. It soon turns out, however, that the segment (number between 1 and 10), the VIP status (a Boolean indicating if the client is a VIP client or not), and the creditscore (A,B,C, or D) are changing very often for the clients, leading to a very large client dimension in which the addresses, names, and date-of-births of the clients are unnecessarily often repeated due to the rapid changes in the attributes segment, VIP, and creditscore.

   (a) Propose a change to the schema that would still allow to store segment, VIP, and creditscore for the clients but without the unnecessary repetitions of the other attributes of the clients. The schema should still allow for rolling up on segment, VIP, and credit-score.

   (b) Illustrate your solution by showing in which relations what tuples would be inserted in the following situation: a client makes two purchases. In-between these two purchases the client's vip status and segment changes.