Midterm for the Course Datawarehousing, SS 2002

Prof. R. Bayer, Institut für Informatik, TUM
June 14,2002

Task 1: Schema design

Assume that the Technical University wants to build a datawarehouse for the grades given by Professors to Students in examinations for Lectures. There are 3 dimensions for ***Lectures***, ***Students*** and ***Professors.*** The fact table ***Grades*** contains the measure **grade** encoded as **smallint.** The dimensions have the hierarchies given below; keys and attribute types are in bold fonts:

Lectures

year	**smallint**	for [1991:2010]
semester	**char(6)**	for ("winter", "summer")
faculty	**char(30)**	for the names of the 13 faculties, the longest being "Landwirtschaft und Gartenbau"
lecture_title	**char(50)**	assume there are at most 100 lecture titles per faculty

Students

matrikel_nr	**integer**	assume that the TU has 20.000 students
stud_last_name	**char(30)**	
stud_first_name	**char(20)**	
semester	**smallint**	

Professors

faculty	**char(30)**	
prof_last_name	**char(30)**	assume there are at most 50 Professors with unique last names per faculty

Question 1: (7 points) What is the schema and the composite key for the fact table **Grades**?

Question 2: (3 points) What is the size of the key of **Grades**, if the above data types are used?

Question 3: (5 points) In addition to the assumptions stated on page 1 assume that every student takes 5 examinations per year,

- indicate below the cardinality of each hierarchy level
- indicate below the number of bits needed for the surrogate encoding of each hierarchy level
- what is the cardinality of the base cube?
- what is the cardinality of occupied cells?
- what is the sparsity of the datacube?

		Cardinality	**Number of bits for surrogate**
Lectures			
year	**smallint**		
semester	**char(6)**		
faculty	**char(30)**		
lecture_title	**char(50)**		
Students			
matrikel_nr	**integer**		
stud_last_name	**char(30)**		
stud_first_name	**char(20)**		
Professors			
faculty	**char(30)**		
prof_last_name	**char(30)**		
prof_first_name	**char(20)**		

Task 2: Surrogates

Question 4: (2 points) For each dimension determine the number of bits needed for the compound surrogate.

Question 5: (2 points) How many bytes are required for the composite key of the fact table **Grades_surr** if compound surrogates are used instead of the key attributes of **Grades**?

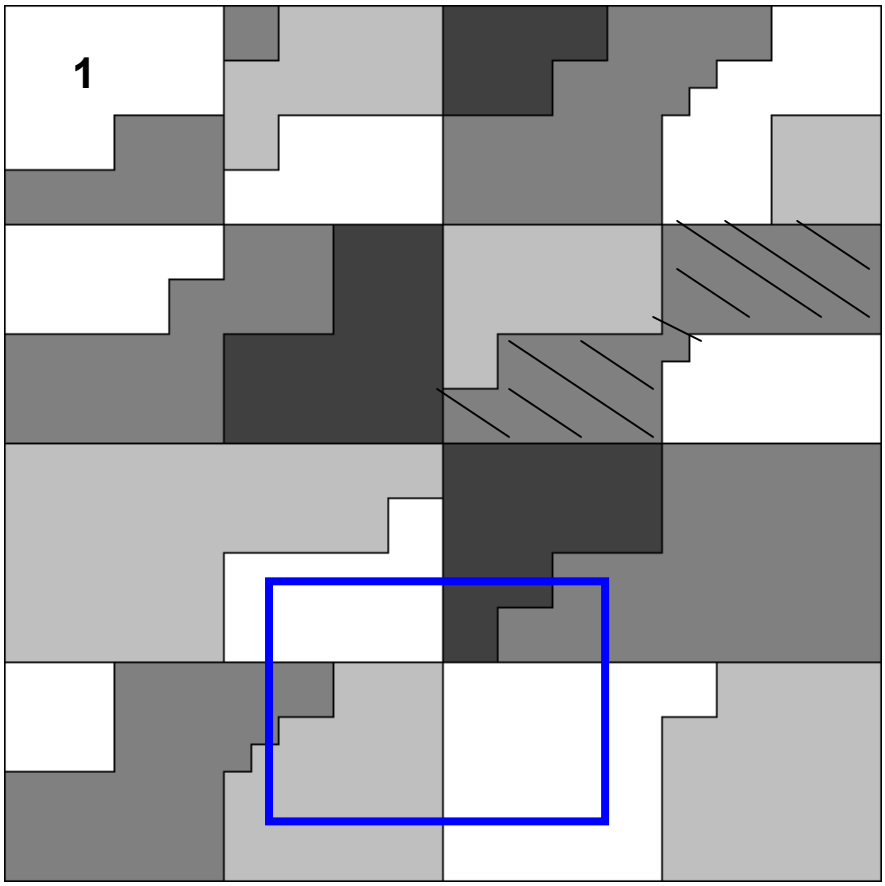
Question 6: (2 points) Compare the size of a tuple of **Grades** with the size of a tuple of **Grades_surr**.

Task 3: UB-trees

Question 7: (3 points) For the following partitioning of space by 24 connected (i.e. without jump-regions) z-regions, number the z-regions starting from 1 in the upper left corner in z-order.

Question 8: (3 points) Construct the lower and upper address (in dot notation) of the striped region.

Question 9: (1 point) Which regions must be fetched from disk to answer the query with the blue query box?



Task 4: Indexing

Question 10: (2 points) Assuming that you want bit map indexes on the attributes **year** and **semester** of the table **Grades.** How many bit vectors are required?

Question 11: (2 points) What is the length of an uncompressed bit vector?

Question 12: (2 points) What is the total size of the two bit map indexes?

Question 13: (1 point) Does the size of the bit map indexes change, if you want to use them for **Grades_surr** instead of for **Grades?** Justify your answer.

Task 5: Query Formulation

Question 14: (5 points) You want to compute the average grade achieved for every semester and for every student in semester 4 resulting in a table with the structure

matrikel_nr, stud_first_name, stud_last_name, year, semester, avg(grade)

Write the SQL query for the schema for **Grades** to carry out this analysis.