Exercises for the Course Datawarehousing, SS 2002, Prof. R. Bayer, TUM

Exercise Sheet 2

Exercise 4, Ch.2-5: Schema optimization

Consider the data warehouse schema of Exercise 1. Reduce the dimensionality of the fact table by exploiting the fact that the location of a mobile phone user usually can only be determined by the current cell. Make sure that the location information of the cell (specified by the location of the cell tower) is not lost.

a) Define the new schema

b) How many dimension tables are now required?

c) What is the size of the new universe, assuming around 20.000 cells in Germany?

Exercise 5, Ch.2-5: Schema optimization

Are there further possibilities to reduce the dimensionality or the size of the fact table without losing information? If yes, present the new schema and calculate the new size of the universe and the fact table with data of one year. What is the disadvantage (if there is any) of this optimization?

Exercise 6, Ch.2-5: Schema extension

Extend the schema of Exercise 5 to support following location-based analysis: "Average percentage of calls within the city (or region, state) where the customer is living". Provide an extended star-schema as well as a snow-flake schema. Give a rough estimate of how much space can be saved with the snow-flake schema.

Exercises for the Course Datawarehousing, SS 2002, Prof. R. Bayer, TUM

Solutions for Exercise Sheet 2

Exercise 4, Ch.2-5: Schema optimization

Consider the data warehouse schema of Exercise 1. Reduce the dimensionality of the fact table by exploiting the fact that the location of a mobile phone user usually can only be determined by the current cell. Make sure that the location information of the cell (specified by the location of the cell tower) is not lost.

a) Define the new schema

CELL(int cell_id, int X, int Y)

FACT(caller, callee, int caller_cell, int callee_cell, int duration)

b) How many dimension tables are now required? **4 (Caller, Callee TimeofCall, CELL) or 3 by combining Caller and Callee → Customer**

c) What is the size of the new universe, assuming around 20.000 cells in Germany?

$10*12*31*24*60 * 2*10^4 * 2*10^4 * 100 * 10^7 * 100*10^7 = 2{,}2 * 10^{33}$

Exercises for the Course Datawarehousing, SS 2002, Prof. R. Bayer, TUM

Solutions for Exercise Sheet 2

Exercise 5, Ch.2-5: Schema optimization

Are there further possibilities to reduce the dimensionality or the size of the fact table without losing information? If yes, present the new schema and calculate the new size of the universe and the fact table with data of one year.

FACT(int time_id, int caller, int callee, int cell_caller, int cell_callee, int duration)
TIME(int time_id, small int year, string month, small int day, small int hour, small int minute)
CUSTOMER(int customer_id, small int prefix, int number, ….)
CELL(int cell_id, int X, int Y)

Universe size = same
FACT table: # tuples * 24B ➔ ½ of original fact table → saves 1 TB per year !!

Disadvantage: higher processing cost to get missing fields if they are required by a query, e.g., grouping over months, years. So-called "residual joins" required

Exercises for the Course Datawarehousing, SS 2002, Prof. R. Bayer, TUM

Solutions for Exercise Sheet 2

Exercise 6, Ch.2-5: Schema extension

Extend the schema of Exercise 5 to support following location-based analysis: "Average percentage of calls within the city (or region, state) where the customer is living". Provide an extended star-schema as well as a snow-flake schema. Give a rough estimate of how much space can be saved with the snow-flake schema.

Star:
CUSTOMER(int customer_id, smallint area_code, int tel_number, char(40) Country, char(40) State, char (40) Region, char(40) City)
CELL(int cell_id, int X, int Y, char(40) Country, char(40) State, char (40) Region, char(40) City)

Snowflake: factoring out of the "shared hierarchy"
Geography(**int city_id,** char(40) Country, char(40) State, char (40) Region, char(40) City)
CUSTOMER(int customer_id, smallint area_code, int number, **int city_id**)
CELL(int cell_id, int X, int Y, **int city_id**)

Savings: 156B per tuples → 4,6 GB for customer dimension assuming 30 Mio customers
→ 36 MB for CELL assuming 200K cells