ESF
EIROPAS SOCIĀLAIS FONDS
EIROPAS SAVIENĪBA
IEGULDĪJUMS TAVĀ NĀKOTNĒ

This work has been supported
by ESF project No.
2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044

Query-Driven Method for Improvement of Data Warehouse Conceptual Model

Darja Solodovnikova, Laila Niedrite, Aivars Niedritis

University of Latvia

International Conference on Information Systems Development
2012, Prato, Italy

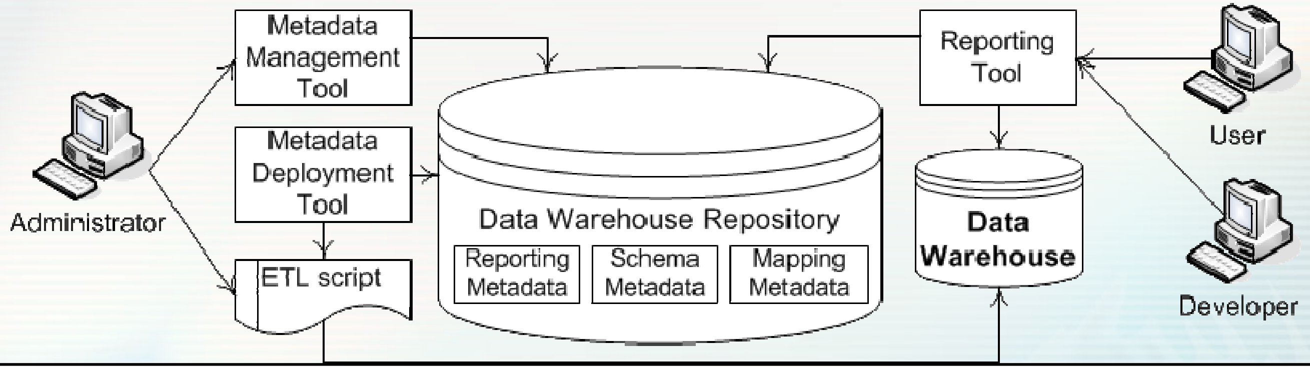
Multidimensional Model of Data Warehouses

- Data warehouses (DW) are based on multidimensional models, which contain:
  - facts (the goal of the analysis),
  - measures (quantitative data),
  - dimensions (qualifying data),
  - dimension attributes that form classification hierarchies.

- Approaches to the development of conceptual models of DW :
  - supply driven (also known as data-driven)
  - demand driven
    - named according to the requirements elicitation method,
    - user-driven, process-driven , goal-driven etc.

Data Warehouse Evolution

- Data warehouse conceptual models tend to evolve, because of changes in :
  - data sources of DW
  - information requirements of users
- In our approach:
  - a data warehouse is a part of the data warehouse evolution framework
  - we use metadata about existing data warehouse schema, mappings of its elements to data source elements, as well as existing reports defined on the data warehouse schema



Research problem and our proposal

- The problem addressed in our research is:
    - how to reconcile new information requirements with the existing conceptual model of the data warehouse

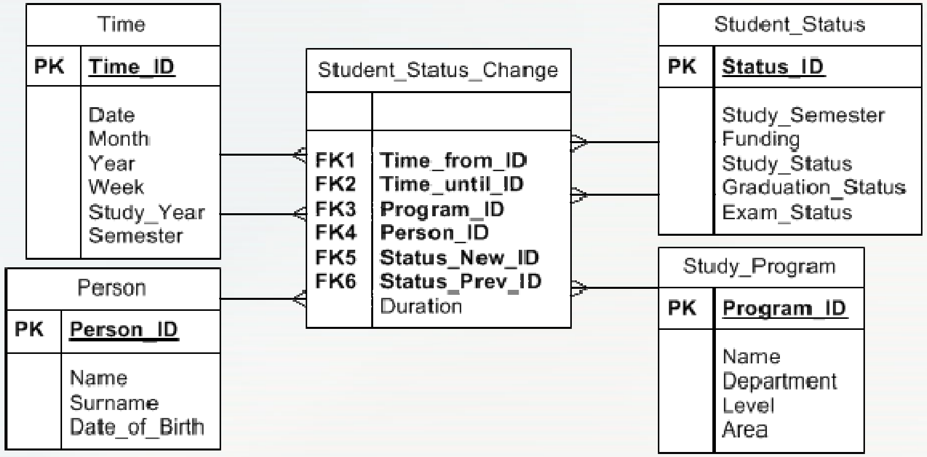
- We propose a new type of demand-driven method:
    - query-driven method
        - elicits the requirements from existing queries on data sources and their usage statistics,
        - combines the needs of users with existing data supply
    - method recommends changes to existing data warehouse schemata
    - new version of DW can be constructed according to the recommendations

Related work

- Our method for eliciting user information requirements presumes that the queries against the source database reflect the interests and needs of users
- Some other works also exploit similar idea:
  - to improve query performance, but not to get a conceptual model,
  - analyze expected queries (documented during interviews) instead of real queries
  - analyze only the structure of implemented queries in source systems, regardless of the real usage of the queries
- The proposed method:
  - We take into account not only structure but also the usage statistics of queries in source systems
  - Our method is tailored to a definite purpose to recommend the necessary changes in existing data warehouse schema

Example

- Data Warehouse – the existing version



Time	
PK	Time_ID
	Date Month Year Week Study_Year Semester

Person	
PK	Person_ID
	Name Surname Date_of_Birth

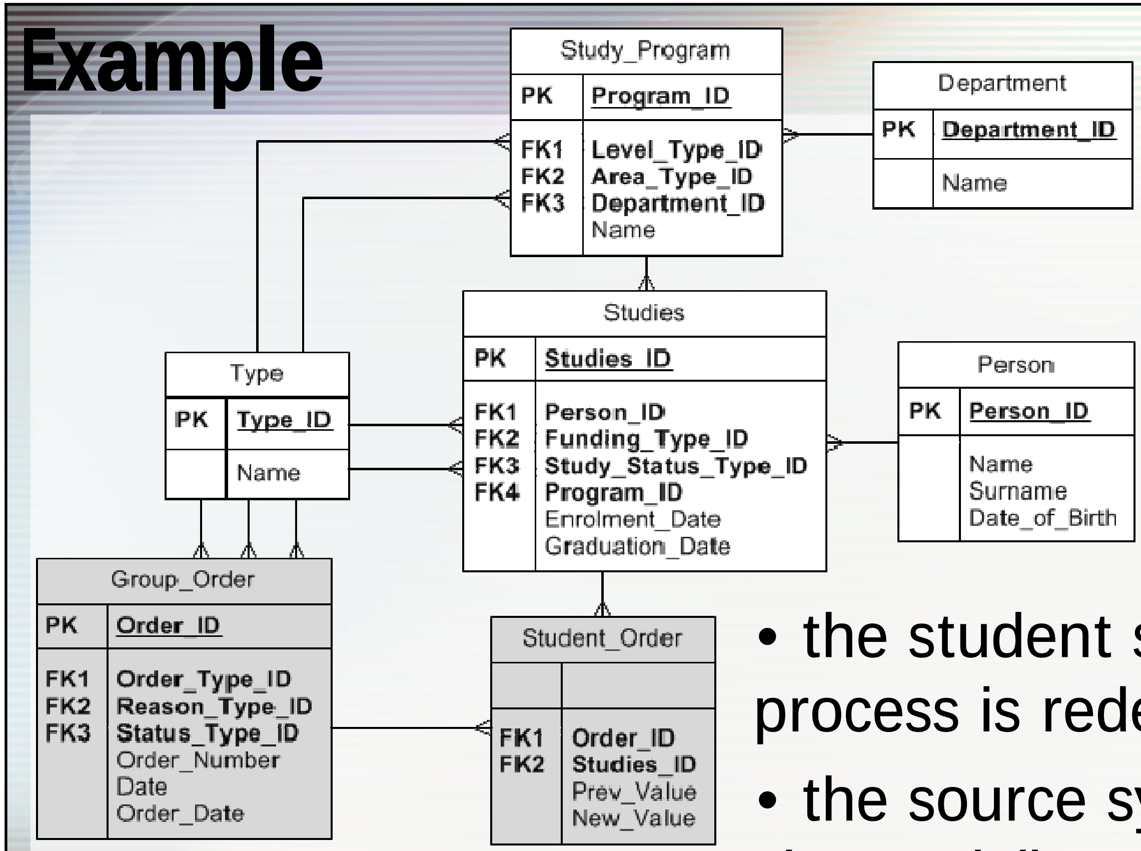
Student_Status_Change	
FK1 FK2 FK3 FK4 FK5 FK6	Time_from_ID Time_until_ID Program_ID Person_ID Status_New_ID Status_Prev_ID Duration

Student_Status	
PK	Status_ID
	Study_Semester Funding Study_Status Graduation_Status Exam_Status

Study_Program	
PK	Program_ID
	Name Department Level Area

- The data is obtained from the university data management system.

Example



Study_Program
PK Program_ID
FK1 Level_Type_ID
FK2 Area_Type_ID
FK3 Department_ID
Name

Department
PK Department_ID
Name

Studies
PK Studies_ID
FK1 Person_ID
FK2 Funding_Type_ID
FK3 Study_Status_Type_ID
FK4 Program_ID
Enrolment_Date
Graduation_Date

Type
PK Type_ID
Name

Person
PK Person_ID
Name
Surname
Date_of_Birth

Group_Order
PK Order_ID
FK1 Order_Type_ID
FK2 Reason_Type_ID
FK3 Status_Type_ID
Order_Number
Date
Order_Date

Student_Order
FK1 Order_ID
FK2 Studies_ID
Prev_Value
New_Value

- the student status changing process is redesigned
- the source system model is changed (in grey)

•How the DW model should be changed to reflect the analysis needs of users and new data model of the source system?

Information Sources for Query Usage Study (1)

- We will examine a source system that collects different logs about functions and data usage
- The function usage:
  - Logging is invoked when the function is performed
  - Log format: *Function_name, user, date.*
  - This log can exhibit the most frequently used functions
  - the data manipulation statements that are used within these functions should be analyzed
- The data changes:
  - Each table has its own trigger for logging insert, update and delete statements
  - Log format: *Table_name, user, date, change_type, change_text*
  - The statistics about changes for each table and attribute is computed

Information Sources for Query Usage Study (2)

- A special kind of functions:
  - reports with a wide range of parameters, that user can change
  - The queries in reports represent also the information requirements of users
  - format of such log is the following*:*

  Function_name, user, date, parameter_text,

    - where *parameter_text* contains values in the format: *p1=<value1>... pN=<valueN>*,
    - where *p1...pN* are in the format: *<table_name.column_name>*.

Information Sources for Query Usage Study (3)

- The structure of data manipulation statements used in functions of source systems is examined

- Two kinds of information sources can be prepared.
  - INSERT and UPDATE statements :
    - *Function_name, Action_Type, List_of_columns, Where_condition*,
  - SELECT statements :
    - *Function_name, Joins, Where_condition, Group_By, Columns_Select, Aggregation,*

Query-Driven Method (1.step)

- Pre-processing of Information Sources
  - two usage tables are constructed
    - **data modification table** - contains data about the frequency of INSERT and UPDATE statements performed by functions with tables
      - *Function_name, List_of_columns, Where_condition, count.*
    - **data selection table** - contains data about the frequency of SELECT statements performed by functions with tables
      - *Function_name, Joins, Where_condition, Group_By, Columns_Select, Aggregation, count.*

Query-Driven Method (2.step)

- Analysis of Data Modification Statements
  - data modification table is analysed*.*

  - For each column, we calculate the number of times it was updated or inserted
    - we summarize *count* for every occurrence of the column in ***List_of_columns*** of the data modification table
    - we choose the Top-N columns with the biggest number of times they were updated or inserted and consider such columns as **potential measures*.***

  - we identify the **potential dimension attributes** from the Top-N columns with the biggest number of times they were used in ***Where_condition*** in the data modification table.

  - We assume that **potential fact measures could be analysed together with potential dimension attributes** if they are used together in one data modification statement.

Query-Driven Method (3.step)

- Analysis of Data Selection Statements
  - the Top-N most frequently used columns in *Where_condition* or *Group_By* are considered as **potential dimension attributes**.
  - The Top-N columns with the biggest number of times they were used in *Aggregation* are considered the **potential measures**.
  - Several potential measures belonging to one source table are united into a potential data warehouse **fact table**,
  - Several potential attributes belonging to one source table are united into a potential data warehouse **dimension table**

Query-Driven Method (4.step)

- Conflict Resolution
  - it is possible that the same source column can be identified as potential fact measure **and** as a potential dimension attribute
  - In such case we analyse the data type of the column:
    - If the data type of the column is **numeric**, then we consider such column as a **measure,**
    - If the data type of the column is **character** or **date**, then we refer to such column as to a **dimension attribute**

Query-Driven Method (5.step)

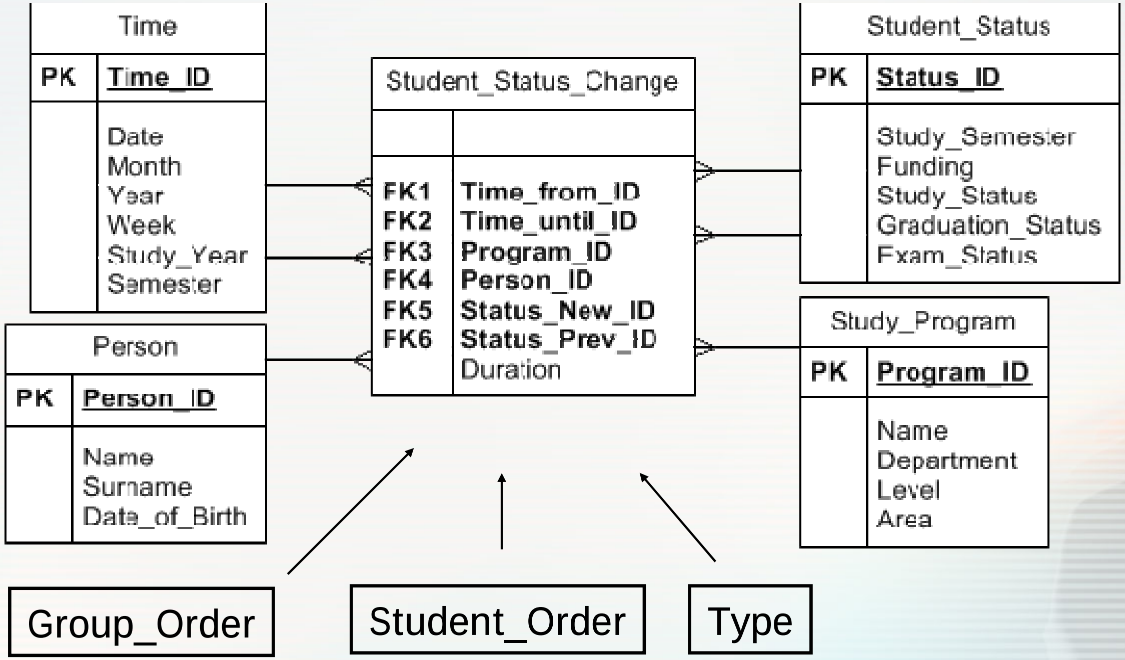
- Generation of Potential Improvements:
  - the **potential** data warehouse schema elements are compared with the **existing** data warehouse schema
    - Mappings between source table columns and existing data warehouse elements are used in this comparison.
  - Non-existing potential data warehouse schema elements are recommended to be added to the existing data warehouse schema.
  - Schema changes supported by the data warehouse evolution framework:
    - add a new dimension attribute or fact measure,
    - add a new dimension or fact table,
    - connect dimension to a fact table,
    - add a new dimension hierarchy or new hierarchy level.
  - Recommended changes should be accepted by the DW administrator

Example (1)

- The proposed method was applied to the student status change data warehouse and student management system
- Student status changing process was logged and analyzed:
  - The following columns are encountered frequently in *Where_Condition, Group_By* and *Columns_Select* of the data selection table:
    - Attribute **Name** of the table **Type**;
    - **Order_Date** and **Order_Number** of the table **Group_Order**;
    - **Prev_Value** and **New_Value** of the table **Student_Order**.
  - these columns are considered as potential dimension attributes

Example (2)

- The proposed method recommends to extend the data warehouse schema with the dimensions Group_Order, Student_Order and Type.



Time

PK	Time_ID
	Date Month Year Week Study_Year Semester

Student_Status_Change

FK1 FK2 FK3 FK4 FK5 FK6	Time_from_ID Time_until_ID Program_ID Person_ID Status_New_ID Status_Prev_ID Duration

Student_Status

PK	Status_ID
	Study_Semester Funding Study_Status Graduation_Status Exam_Status

Person

PK	Person_ID
	Name Surname Date_of_Birth

Study_Program

PK	Program_ID
	Name Department Level Area

Group_Order

Student_Order

Type

Example (3)

- After the **manual revision** of recommended improvements:
  - it was decided to unite dimensions Group_Order, Student_Order and Type in **one dimension Order** with the following attributes:
    - Order_Date, Order_Number, Prev_Value, New_Value.
  - the column Name of the table Type was added as 3 attributes of the dimension Order:
    - Order_Type, Reason_Type and Status_Type.

Conclusions and Future Work

- The proposed method:
  - allows to keep track of actual analysis tasks performed in operational data sources
  - adjust the existing data warehouse schema according to the most frequent queries.
  - the method is used to recommend necessary changes for a new data warehouse schema version.
  - a data warehouse schema also serves as a representation of the existing data warehouse analysis needs.
- Future work:
  - We are working on implementation of the method in more complicated cases and evaluation of constructed models to be convinced that the new version of the data warehouse is adjusted to real analysis needs

Thank you!