Evolution-Oriented User-Centric Data Warehouse

Darja Solodovnikova and Laila Niedrite

University of Latvia

19th International Conference on Information Systems Development
August 25-27, 2010, Prague, Czech Republic

EIROPAS SOCIĀLAIS FONDS

This work has been supported by ESF projects No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044 and 2009/0138/1DP/1.1.2.1.2/09/IPIA/VIAA/004

Outline

- Data warehouse evolution
- Motivating example
- Data warehouse evolution framework
- Metadata repository
  - Logical metadata
  - Physical metadata
  - Semantic metadata
  - Reporting metadata
- Reports on multiversion data warehouse
  - Queries on multiple versions
  - Hierarchy versions
  - Term versions
- Conclusions and future work

Data Warehouse Evolution and Effects

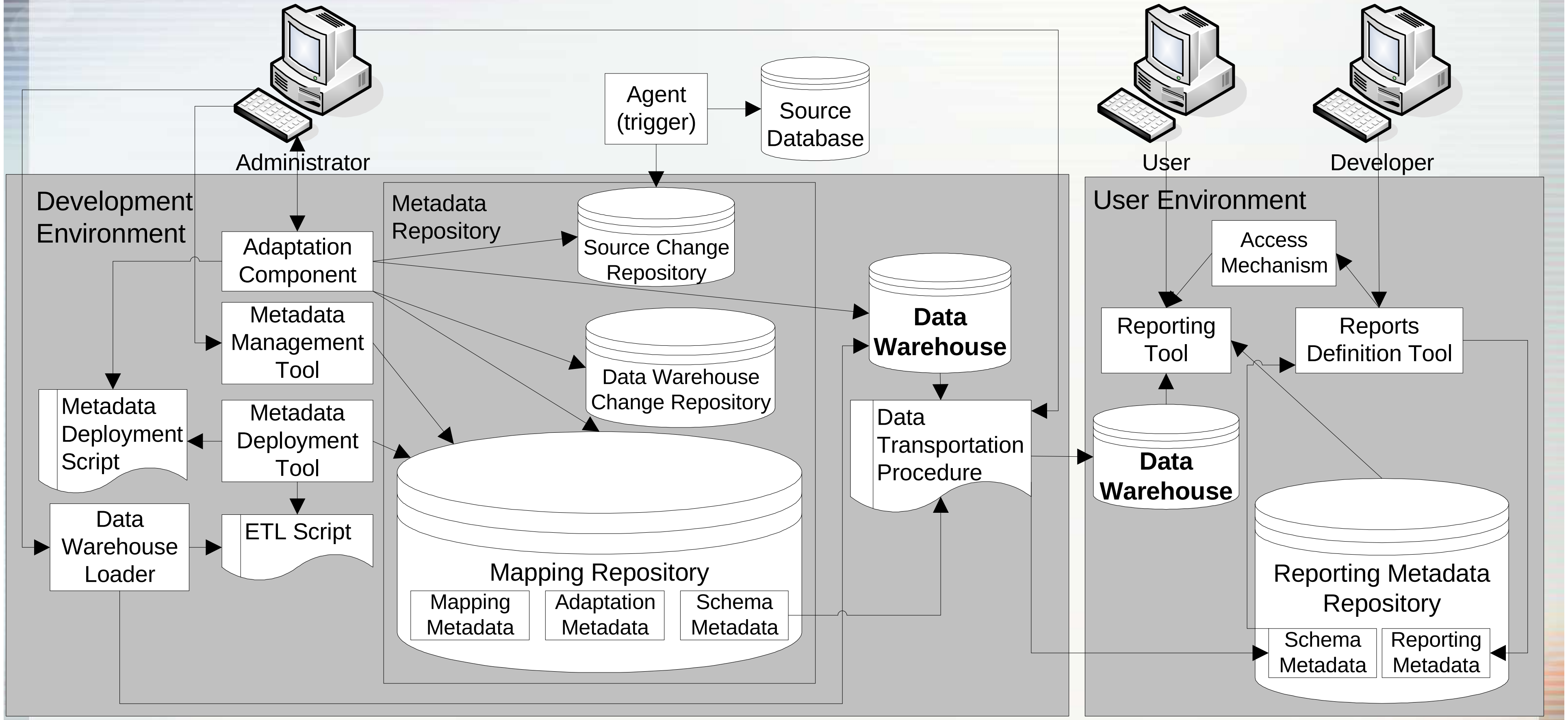
- Changes
  - Data sources
  - Business requirements
- Effects
  - Changes in data warehouse schema
  - Invalid data extraction, transformation and loading (ETL) processes
  - Invalid reports on data warehouse schema
- Data warehouse schema versions

Data Warehouse Schema Version

- *Schema version* is a schema that reflects the business requirements during a given time interval, called its validity, that starts upon schema creation and extends until the next version is created

(Golfarelli, M., et. Al, 2006)

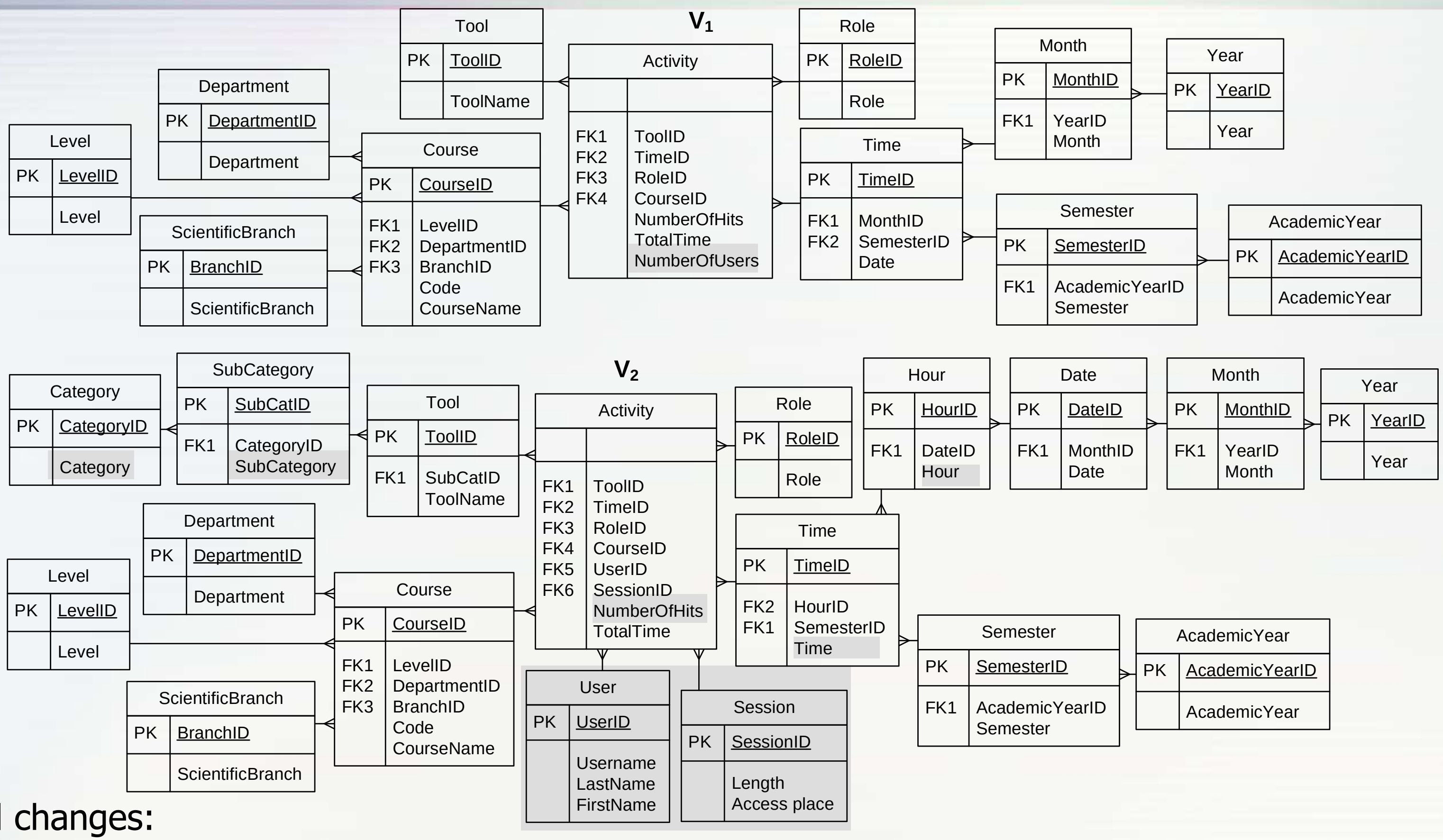
Data Warehouse Evolution Framework



Evolution support:

- **Physical changes** operate with database objects
- **Logical changes** modify schema metadata
- **Semantic changes** adapt meanings of data objects

Motivating Example



V1

Tool
PK ToolID
ToolName

Activity
FK1 ToolID
FK2 TimeID
FK3 RoleID
FK4 CourseID
NumberOfHits
TotalTime
NumberOfUsers

Role
PK RoleID
Role

Month
PK MonthID
FK1 YearID
Month

Year
PK YearID
Year

Department
PK DepartmentID
Department

Level
PK LevelID
Level

Course
PK CourseID
FK1 LevelID
FK2 DepartmentID
FK3 BranchID
Code
CourseName

ScientificBranch
PK BranchID
ScientificBranch

Time
PK TimeID
FK1 MonthID
FK2 SemesterID
Date

Semester
PK SemesterID
FK1 AcademicYearID
Semester

AcademicYear
PK AcademicYearID
AcademicYear

V2

Category
PK CategoryID
Category

SubCategory
PK SubCatID
FK1 CategoryID
SubCategory

Tool
PK ToolID
FK1 SubCatID
ToolName

Activity
FK1 ToolID
FK2 TimeID
FK3 RoleID
FK4 CourseID
FK5 UserID
FK6 SessionID
NumberOfHits
TotalTime

Role
PK RoleID
Role

Hour
PK HourID
FK1 DateID
Hour

Date
PK DateID
FK1 MonthID
Date

Month
PK MonthID
FK1 YearID
Month

Year
PK YearID
Year

Department
PK DepartmentID
Department

Level
PK LevelID
Level

Course
PK CourseID
FK1 LevelID
FK2 DepartmentID
FK3 BranchID
Code
CourseName

ScientificBranch
PK BranchID
ScientificBranch

Time
PK TimeID
FK2 HourID
FK1 SemesterID
Time

Semester
PK SemesterID
FK1 AcademicYearID
Semester

AcademicYear
PK AcademicYearID
AcademicYear

User
PK UserID
Username
LastName
FirstName

Session
PK SessionID
Length
Access place

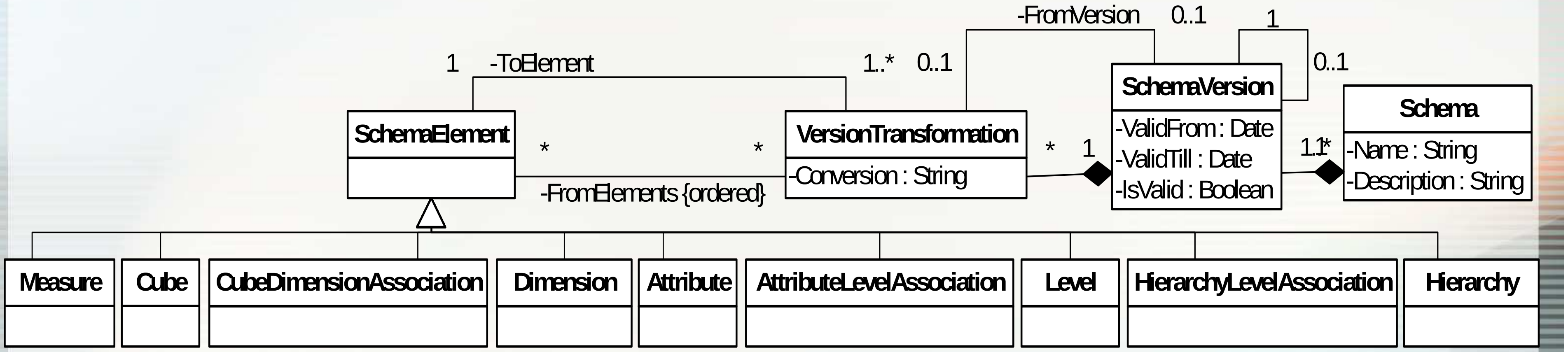
- Physical changes:
  - New dimensions User and Session;
  - New dimension attributes Time and Hour of the Time dimension, SubCategory and Category of the Tool dimension;
  - Deletion of measure NumberOfUsers from the Activity cube.
- Logical changes:
  - Connection of the User and Session dimensions to the Activity cube;
  - New hierarchy ToolHierarchy of the Tool dimension;
  - New levels ToolName, SubCategory and Category of the ToolHierarchy; levels Time and Hour of the TimeHierarchy;
  - Connection of attributes ToolName, SubCategory, Category, Time and Hour to the appropriate hierarchy levels.
- Semantic change:
  - Change of meaning of the measure TotalTime from ‘Usage time in hours‘ to ‘Usage time in seconds’.

Metadata Repository

- Data warehouse metadata:
  - – Physical metadata
    - Implementation of a data warehouse in RDBMS
  - – Logical metadata
    - Schema versions of a multidimensional data warehouse
  - – Semantic metadata
    - Description of data warehouse elements in business language
- Reporting metadata:
    - Structure of reports
- Common Warehouse Metamodel (CWM) was used as a basis of the proposed metamodel of multidimensional data warehouse.

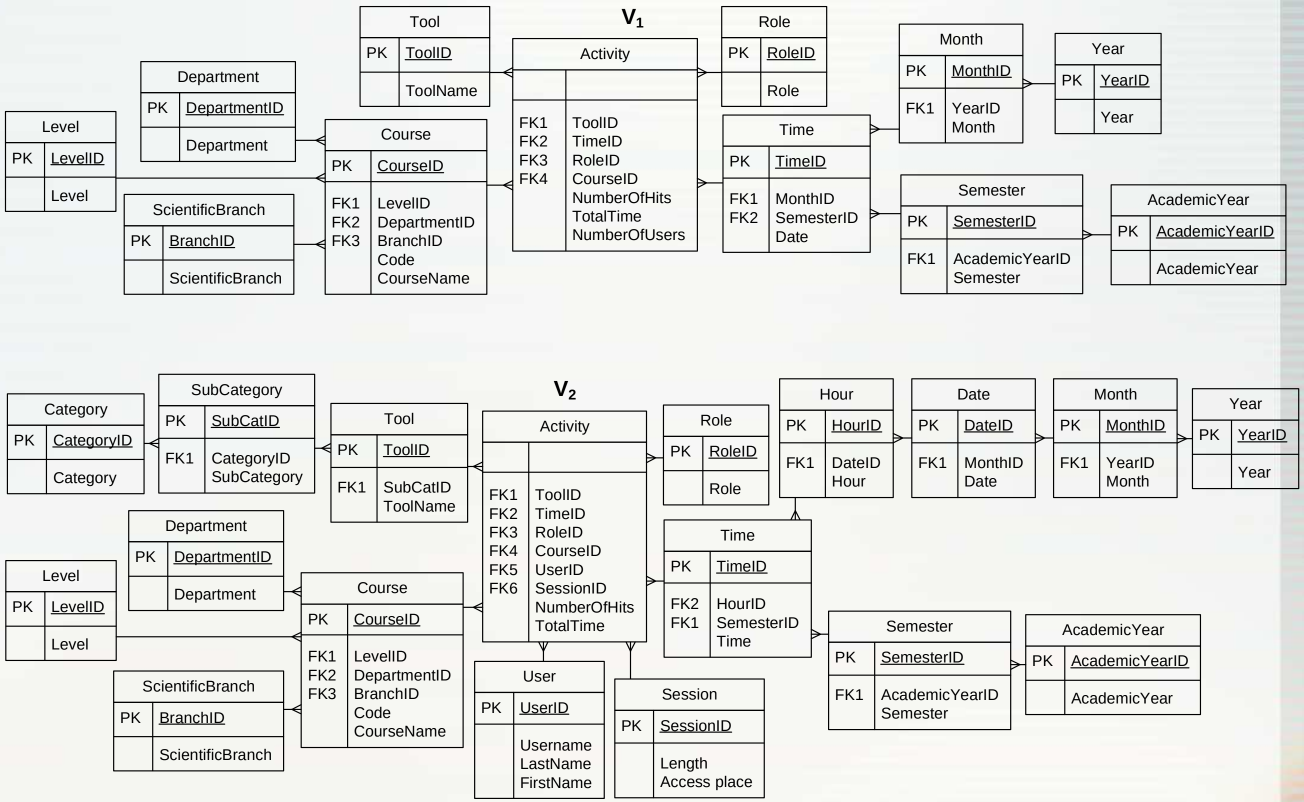
Logical Metadata

- Based on the OLAP package of CWM;
- Contain the main objects of data warehouse, such as dimensions and cubes connected by cube-dimension associations, measures, attributes, hierarchies, etc.



Logical Metadata – Example

- Two schema versions V1 and V2.
- Common elements are connected to versions V1 and V2.
- Measure NumberOfUsers is connected only to version V1.
- Two different measure objects are constructed for each of the measures NumberOfHits and TotalTime and connected to each of the versions.
- Dimensions User and Session and the corresponding cube-dimension associations and cube Activity are connected only to version V2.
- Also the new attributes of the dimensions Time and Tool and corresponding hierarchies exist only in the new version V2.



V_1

Tool
PK ToolID
ToolName

Activity
FK1 ToolID
FK2 TimeID
FK3 RoleID
FK4 CourseID
NumberOfHits
TotalTime
NumberOfUsers

Role
PK RoleID
Role

Month
PK MonthID
FK1 YearID
Month

Year
PK YearID
Year

Department
PK DepartmentID
Department

Level
PK LevelID
Level

Course
PK CourseID
FK1 LevelID
FK2 DepartmentID
FK3 BranchID
Code
CourseName

ScientificBranch
PK BranchID
ScientificBranch

Time
PK TimeID
FK1 MonthID
FK2 SemesterID
Date

Semester
PK SemesterID
FK1 AcademicYearID
Semester

AcademicYear
PK AcademicYearID
AcademicYear

V_2

Category
PK CategoryID
Category

SubCategory
PK SubCatID
FK1 CategoryID
SubCategory

Tool
PK ToolID
FK1 SubCatID
ToolName

Activity
FK1 ToolID
FK2 TimeID
FK3 RoleID
FK4 CourseID
FK5 UserID
FK6 SessionID
NumberOfHits
TotalTime

Role
PK RoleID
Role

Hour
PK HourID
FK1 DateID
Hour

Date
PK DateID
FK1 MonthID
Date

Month
PK MonthID
FK1 YearID
Month

Year
PK YearID
Year

Department
PK DepartmentID
Department

Level
PK LevelID
Level

Course
PK CourseID
FK1 LevelID
FK2 DepartmentID
FK3 BranchID
Code
CourseName

ScientificBranch
PK BranchID
ScientificBranch

Time
PK TimeID
FK2 HourID
FK1 SemesterID
Time

User
PK UserID
Username
LastName
FirstName

Session
PK SessionID
Length
Access place

Semester
PK SemesterID
FK1 AcademicYearID
Semester

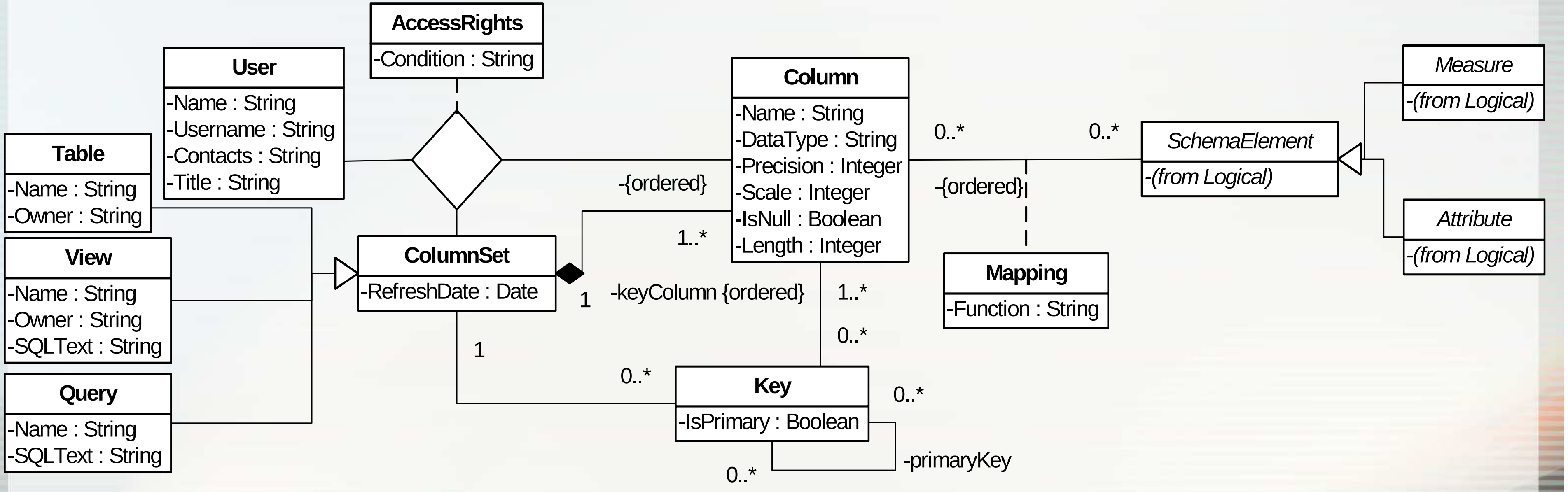
AcademicYear
PK AcademicYearID
AcademicYear

Version transformations:

FromVersion	ToVersion	ToElement	Conversion
V_2	V_1	NumberOfUsers	COUNT(DISTINCT Activity.UserID)
V_2	V_1	NumberOfHitsV1	SUM(Activity.NumberOfHitsV2)
V_2	V_1	TotalTimeV1	SUM(Activity.TotalTimeV2/3600)

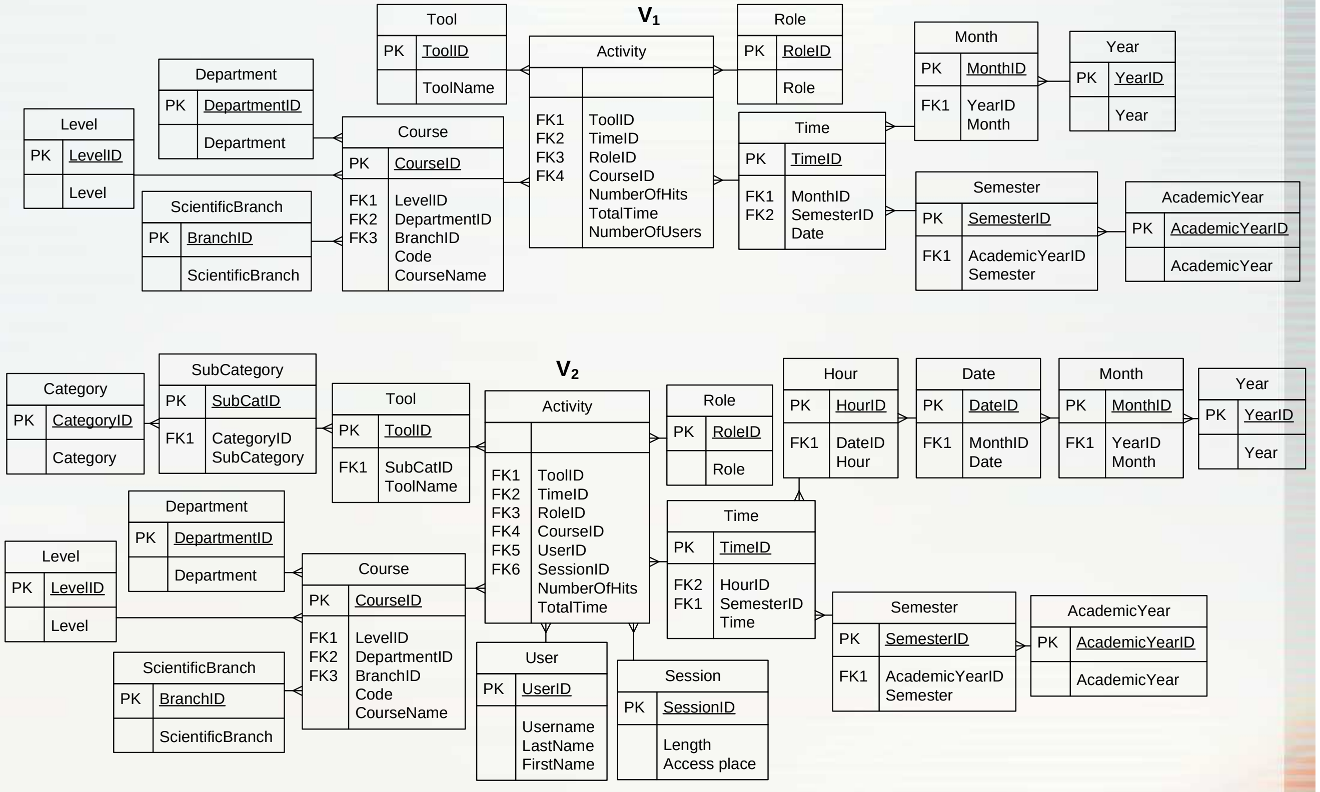
Physical Metadata

- Based on the Relational package of CWM



Physical Metadata & Database – Example

- Schema versions are stored in one physical schema.
- New tables User and Session are created.
- New fictive records with data “All together” are created with an identifier U and S respectively.
- User_ID and Session_ID are added to the table Activity and filled with U and S for all existing data.
- Time and Tool are supplemented with columns Time, Hour and Subcategory, Category.



V_1

Tool: PK ToolID; ToolName

Activity: FK1 ToolID, FK2 TimeID, FK3 RoleID, FK4 CourseID; NumberOfHits, TotalTime, NumberOfUsers

Role: PK RoleID; Role

Month: PK MonthID; FK1 YearID; Month

Year: PK YearID; Year

Department: PK DepartmentID; Department

Level: PK LevelID; Level

Course: PK CourseID; FK1 LevelID, FK2 DepartmentID, FK3 BranchID; Code, CourseName

Time: PK TimeID; FK1 MonthID, FK2 SemesterID; Date

ScientificBranch: PK BranchID; ScientificBranch

Semester: PK SemesterID; FK1 AcademicYearID; Semester

AcademicYear: PK AcademicYearID; AcademicYear

V_2

Category: PK CategoryID; Category

SubCategory: PK SubCatID; FK1 CategoryID; SubCategory

Tool: PK ToolID; FK1 SubCatID; ToolName

Activity: FK1 ToolID, FK2 TimeID, FK3 RoleID, FK4 CourseID, FK5 UserID, FK6 SessionID; NumberOfHits, TotalTime

Role: PK RoleID; Role

Hour: PK HourID; FK1 DateID; Hour

Date: PK DateID; FK1 MonthID; Date

Month: PK MonthID; FK1 YearID; Month

Year: PK YearID; Year

Department: PK DepartmentID; Department

Level: PK LevelID; Level

Course: PK CourseID; FK1 LevelID, FK2 DepartmentID, FK3 BranchID; Code, CourseName

Time: PK TimeID; FK2 HourID, FK1 SemesterID; Time

ScientificBranch: PK BranchID; ScientificBranch

User: PK UserID; Username, LastName, FirstName

Session: PK SessionID; Length, Access place

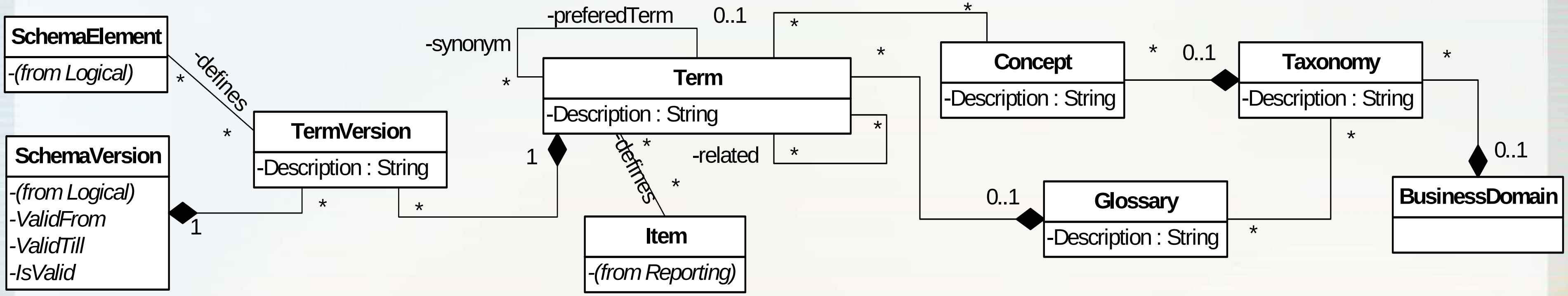
Semester: PK SemesterID; FK1 AcademicYearID; Semester

AcademicYear: PK AcademicYearID; AcademicYear

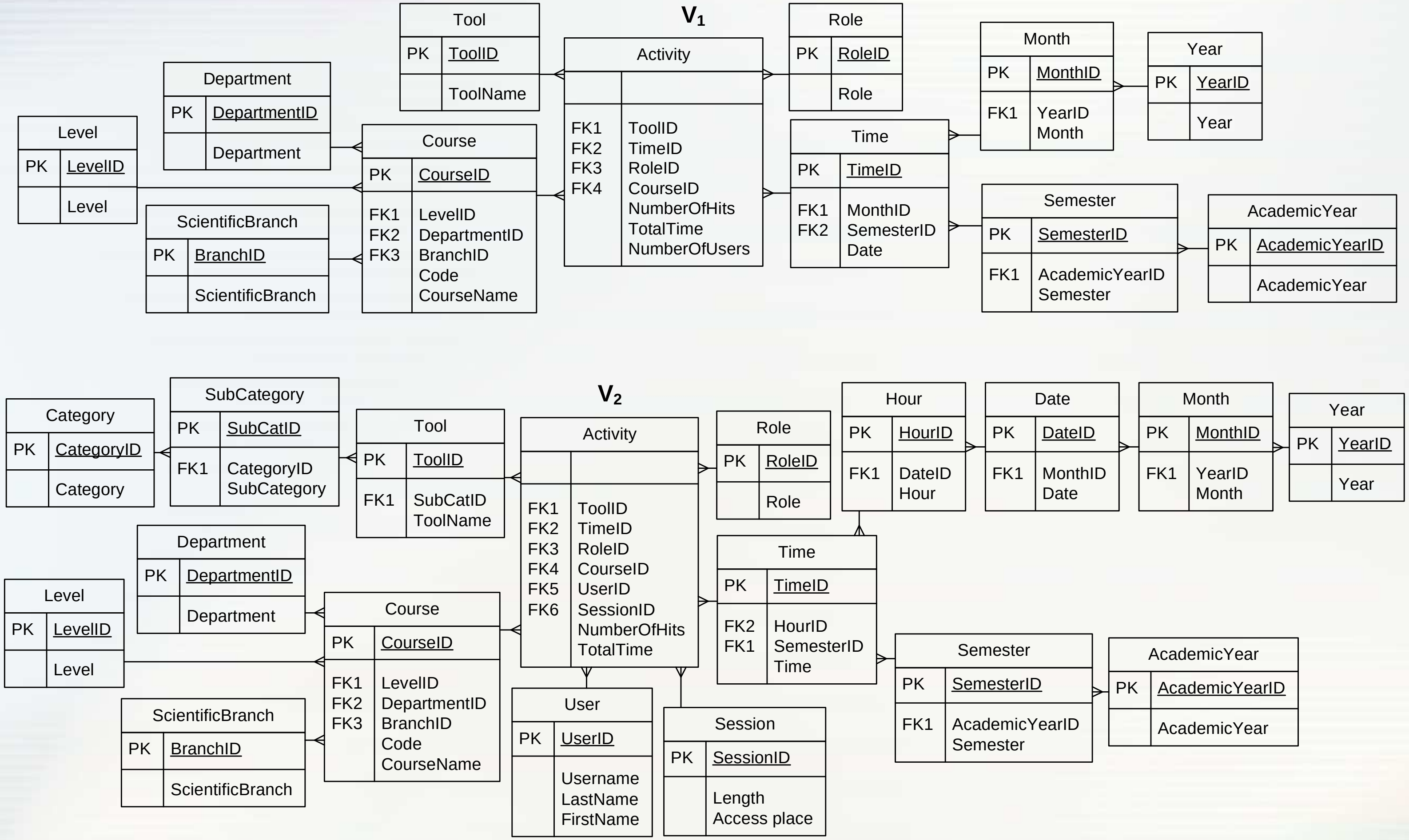
- New columns Subcategory and Category are updated with data for the existing records in tables to form corresponding hierarchies.
- New columns of the Time dimension are updated with fictive data, for example, time ’00:00’.
- NumberOfUsers is not removed, but is no longer updated by ETL processes.
- For the new versions of other measures additional columns are not created.

Semantic Metadata

- Based on the Business Nomenclature package of CWM



Semantic Metadata - Example



V₁

Tool	
PK	ToolID
	ToolName

Activity	
FK1 FK2 FK3 FK4	ToolID TimeID RoleID CourseID NumberOfHits TotalTime NumberOfUsers

Role	
PK	RoleID
	Role

Month	
PK	MonthID
FK1	YearID Month

Year	
PK	YearID
	Year

Department	
PK	DepartmentID
	Department

Level	
PK	LevelID
	Level

Course	
PK	CourseID
FK1 FK2 FK3	LevelID DepartmentID BranchID Code CourseName

ScientificBranch	
PK	BranchID
	ScientificBranch

Time	
PK	TimeID
FK1 FK2	MonthID SemesterID Date

Semester	
PK	SemesterID
FK1	AcademicYearID Semester

AcademicYear	
PK	AcademicYearID
	AcademicYear

V₂

Category	
PK	CategoryID
	Category

SubCategory	
PK	SubCatID
FK1	CategoryID SubCategory

Tool	
PK	ToolID
FK1	SubCatID ToolName

Activity	
FK1 FK2 FK3 FK4 FK5 FK6	ToolID TimeID RoleID CourseID UserID SessionID NumberOfHits TotalTime

Role	
PK	RoleID
	Role

Hour	
PK	HourID
FK1	DateID Hour

Date	
PK	DateID
FK1	MonthID Date

Month	
PK	MonthID
FK1	YearID Month

Year	
PK	YearID
	Year

Department	
PK	DepartmentID
	Department

Level	
PK	LevelID
	Level

Course	
PK	CourseID
FK1 FK2 FK3	LevelID DepartmentID BranchID Code CourseName

ScientificBranch	
PK	BranchID
	ScientificBranch

User	
PK	UserID
	Username LastName FirstName

Session	
PK	SessionID
	Length Access place

Time	
PK	TimeID
FK2 FK1	HourID SemesterID Time

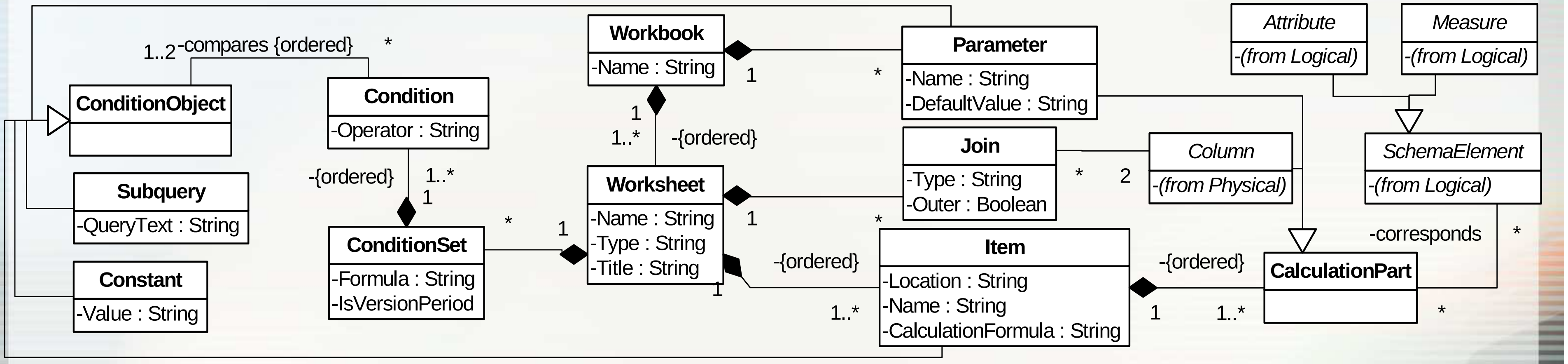
Semester	
PK	SemesterID
FK1	AcademicYearID Semester

AcademicYear	
PK	AcademicYearID
	AcademicYear

- The measure TotalTime is defined by two term versions ‘Total time of all users in hours’ and ‘Total time of each user in seconds’

Reporting Metadata

- CWM contains a package Information Visualization, but these metadata are not sufficient, therefore, a new metamodel was developed.



1..2 -compares {ordered} *

ConditionObject

Subquery
-QueryText : String

Constant
-Value : String

Condition
-Operator : String

-{ordered} 1..* 1

ConditionSet
-Formula : String
-IsVersionPeriod

* 1

Workbook
-Name : String

1 1..* -{ordered}

Worksheet
-Name : String
-Type : String
-Title : String

1 * 1 * 1

-{ordered} 1..*

Parameter
-Name : String
-DefaultValue : String

Join
-Type : String
-Outer : Boolean

* 2

Item
-Location : String
-Name : String
-CalculationFormula : String

-{ordered} 1 1..*

Column
-(from Physical)

Attribute
-(from Logical)

Measure
-(from Logical)

SchemaElement
-(from Logical)

-corresponds *

CalculationPart

*

Reports on Multiversion Data Warehouse I

- Reports are constructed by users, who select desirable terms and term versions from the semantic metadata.
- All necessary reporting metadata are created automatically, according to requirements of users.
- Version Selection
  - If only one version is valid, then report data are presented according to one version;
  - If more than one version is valid, then there are several report presentation options.
  - To determine options of report data presentation, a special *relationship matrix* is used.

Relationships Matrix

- Columns – schema versions
- Rows – schema elements
- Cells:
  - ‘1’, if a schema element exists in a version;
  - ‘2’, if a schema element does not exist in a version, but it is obtainable with the version transformation;
  - ‘0’ otherwise.
- Depending on the values of cells, the following options of report presentation are available:
  - The report can be presented ***in accordance with one particular version***, if all matrix cells of the column, which corresponds to that version, are filled with ‘1’ and all other cells are filled either with ‘1’ or ‘2’. ;
  - If neither of versions does contain all elements, then ***elements from different versions can be presented in one report***, if all matrix cells contain ‘1’ or ‘2’.
  - If any matrix cell contains ‘0’, then the report can only be displayed ***separately for each version***.

Reports on Multiversion Data Warehouse – Example

- *Example report*: Number of students (NumberOfUsers), who used the specific tool (ToolName) in their course during some period of time and the duration of this tool usage (TotalTimeV1).

  Relationship Matrix:

		Version	
		V_1	V_2
Schema Element	CourseName (Course)	1	1
	TotalTimeV1 (Activity)	1	2
	ToolName (Tool)	1	1
	Date (Time)	1	1
	NumberOfUsers (Activity)	1	2

- Example report can be presented according to the first version.

- *Modified report*: Number of students by tool Subcategory.

  Relationship Matrix:

		Version	
		V_1	V_2
Schema Element	CourseName (Course)	1	1
	TotalTimeV1 (Activity)	1	2
	ToolName (Tool)	1	1
	Date (Time)	1	1
	NumberOfUsers (Activity)	1	2
	SubCategory (Tool)	2	1

- Modified report can be presented only with elements from various versions.

Reports on Multiversion Data Warehouse II

- When a user chooses any option of report presentation, an SQL query is built based on reporting metadata and special algorithm:
  1. Analysis of chosen items and determination of used column sets;
  2. Analysis of joins;
  3. Generation of list of conditions;
  4. Grouping and construction of conditions with aggregates functions;
  5. Adding restrictions of user rights;
  6. Simplification and optimization of the query;
  7. Supplementation of a query with version transformations.

Hierarchy Versions

- Hierarchies defined in the logical metadata are used for data analysis.
- Different versions of hierarchies, levels and associations between attributes and hierarchy levels can be created.
- When a user runs a report, all hierarchies and their structure that exist in the particular data warehouse version are identified.
- If elements from different versions are presented in one report, then only hierarchies, which exist in all versions, are available as well as hierarchies that can be transformed by version transformations.

Hierarchy Versions – Example

- *Example report*: Number of students (NumberOfUsers), who used the specific tool (ToolName) in their course during some period of time and the duration of this tool usage (TotalTimeV1).
- Available hierarchies:
  - All Course dimension hierarchies;
  - Time hierarchy, which consists of only three levels Date, Month and Year.

- *Modified report*: Number of students by tool Subcategory.
- Available hierarchies:
  - All Course dimension hierarchies;
  - Time hierarchy, which consists of only three levels Date, Month and Year;
  - Tool hierarchy.

Term Versions

- Term versions are used to separate different meanings of the same schema element.
- If any schema element has multiple term versions, when a user runs a report, he is informed that for the same schema element several term versions exist.
- The user has to choose the preferred term version and then the appropriate schema element version is included in the report.

Term Versions – Example

- *Example report*: Number of students (NumberOfUsers), who used the specific tool (ToolName) in their course during some period of time and the duration of this tool usage (TotalTimeV1).
- Semantics of the measure TotalTime is different in two versions of the data warehouse.
- User must select one of two term versions ‘Total time of all users in hours’ or ‘Total time of each user in seconds’ and the appropriate data are presented in the report.

Conclusions and Future Work

- Approach to data warehouse design:
  - Evolution-oriented
  - User-centric
- Future research:
  - Personalization of reports built on multiple data warehouse versions
  - Support of automatic adaptation of data warehouse according to user needs expressed using terms and term versions from semantic metadata without or with minimal participation of a data warehouse administrator.

Thank you!