Modern Software Engineering Methodologies Meet Data Warehouse Design: 4WD

Matteo Golfarelli

Stefano Rizzi

Elisa Turricchia

University of Bologna - Italy

13th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'11)
August 29, 2011

Summary

- Motivating scenario
- Problems - Goals - Principles
- The methodology: 4WD
- Practical Evidences
- Summary and future work

Motivating scenario

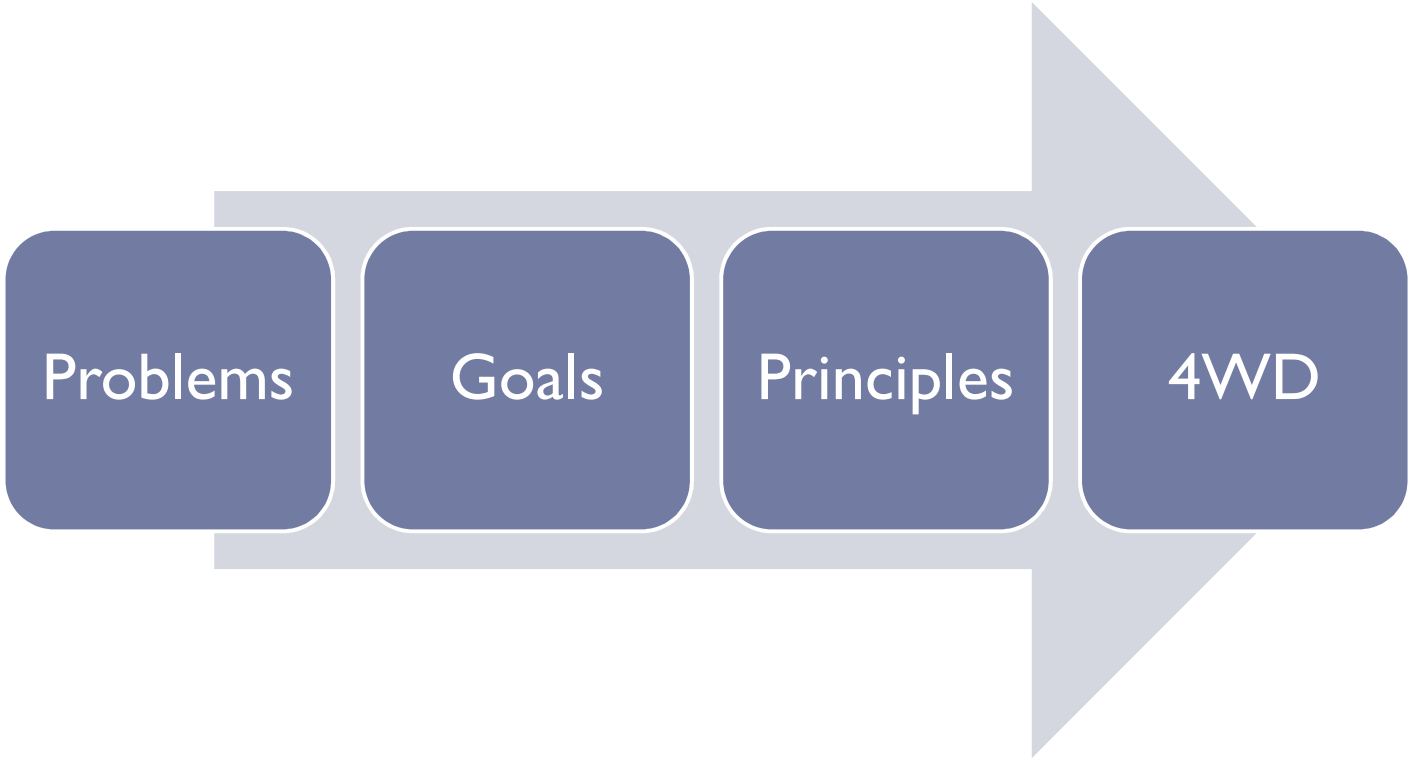
- Data warehouse systems are characterized by a **long** and **expensive** development process that hardly meets the ambitious requirements of today's market
- **Low penetration** of data warehouse systems in small-medium firms
- Data warehouse projects often leave both **customers** and **developers dissatisfied**

Our contribution: ***Four-Wheel-Drive* (4WD)**

An **innovative methodology** to improve the data warehouse development process in terms of **efficiency** and **predictability,** that couples traditional methodologies with Agile approaches

4WD: Research method

- How to design a new methodology for the data warehouse development process?



Problems in the Data Warehouse Development Process

Problem	Motivation
Unclear and uncertain requirements	• Difficult communication between users and developers • Fast business condition evolution • The decision process is flexibly structured and poorly shared across large organization
Long time for delivery	• Data mart centric • Linear development for each data mart
Complexity of a data warehouse	• Data integration • Huge data volume and the workload unpredictability make performance optimization hard

Goals in the Data Warehouse Development Process

Goal	Description	Effect
Reliability	Probability that the delivered system completely and accurately meets user requirements	High-quality and satisfactory final system
Robustness	Capability to quickly react to environment changes	Uncertain and changing requirement management
Productivity	Efficiency of using the resources assigned to the project to speed up system delivery	Shorter and cheaper projects
Timeliness	Accuracy of time and cost assessing	Reliable resource estimates

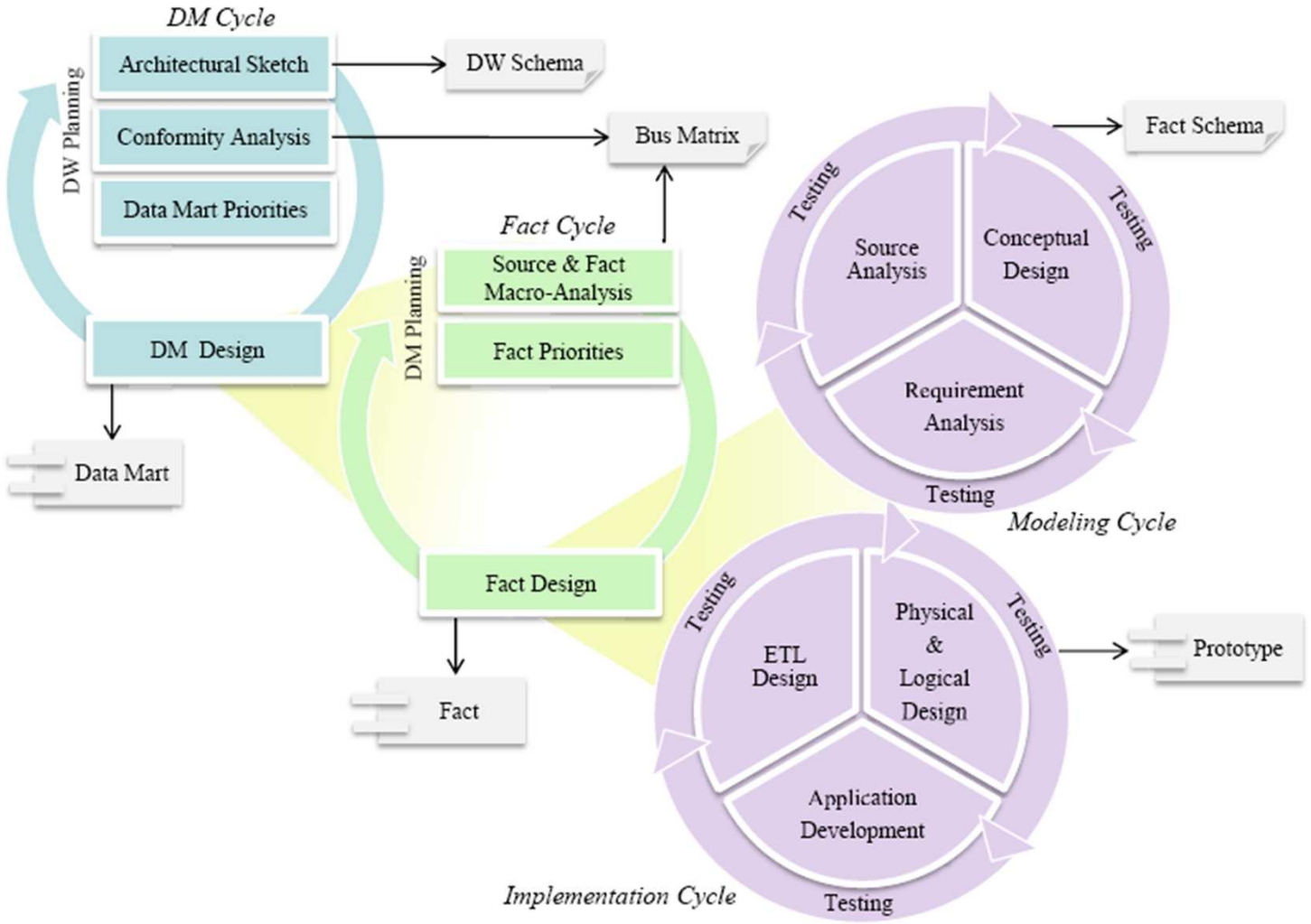
Principles in the Data Warehouse Development Process

Methodologies / Principles	Waterfall [7]	RAD [5]	POSD [6]	SSD [2]	MDA [4]	CBSE [3]	ASD [1]
Incrementality and risk-based iteration		✓		✓			
Prototyping			✓				
User involvement		✓					✓
Component reuse						✓	
Formal and light documentation	✓				✓		✓
Automated schema transformation					✓		

Relationship between **Goals** and **Principles**

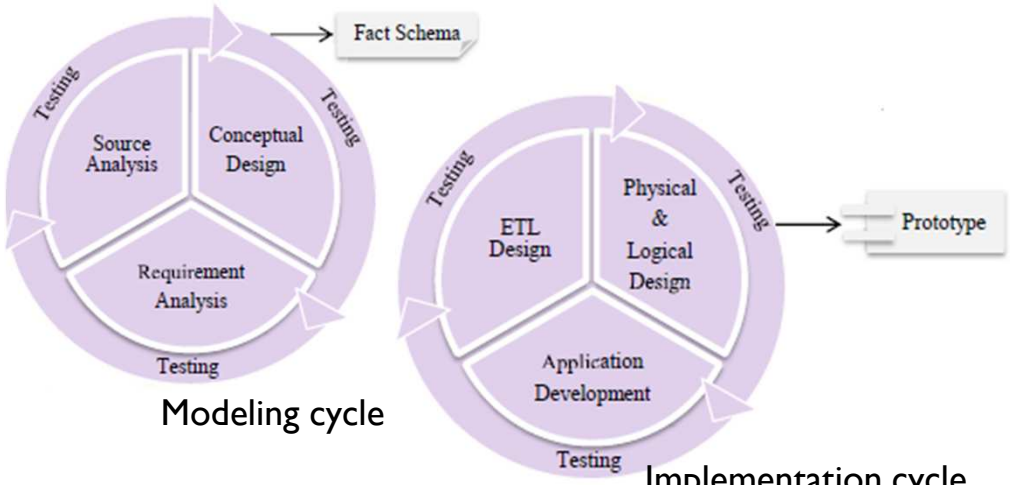
Goals / Principles	Reliability	Robustness	Productivity	Timeliness
Incrementality and risk-based iteration	Continuous feed-back, clearer requirements	Better management of change	Better management of project resources, rapid feedback	Early detection of errors
Prototyping	Frequent tests, easier error detection		Early deliveries	
User involvement	Better requir. validation, better data quality			Early error detection
Component reuse	Error-free components		Faster design	Predictable development
Formal and light documentation	Clearer requirements	Easier evolution	Faster design	
Automated schema transformation	Optimized performances	Easier evolution	Faster design	Predictable design

The Methodology: Four-Wheel-Drive (4WD)

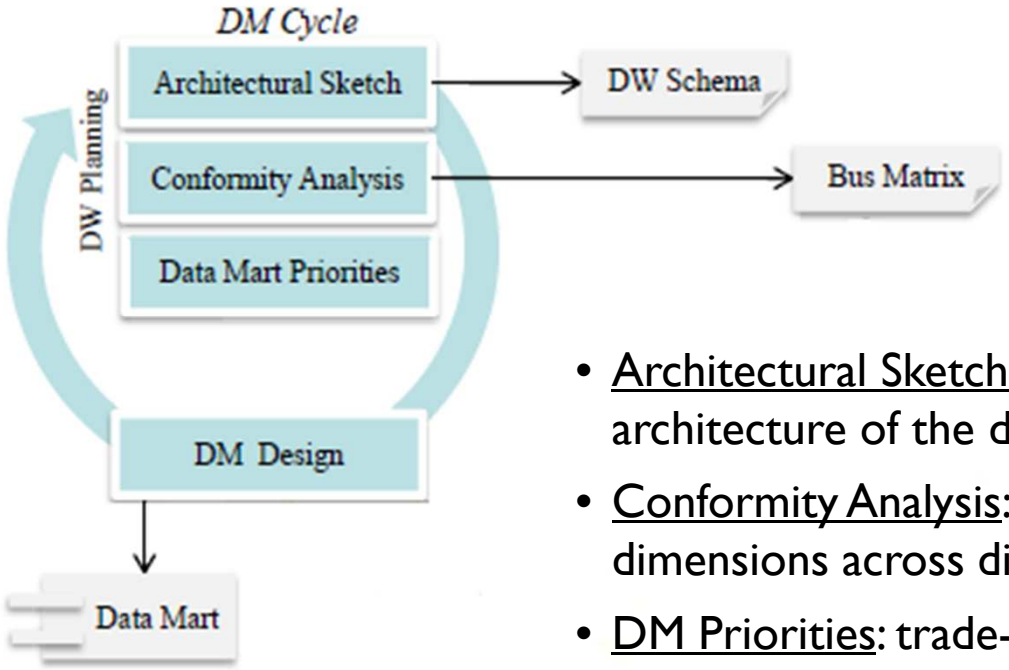


4WD: Description

- Nested iteration cycles:
  - **DM cycle**
    - Global plan for the development of the whole data warehouse
    - Incrementally designs and releases one data mart
  - **Fact cycle**
    - Refines the data mart plan
    - It incrementally designs the facts of a data mart
    - Fact design:
      - **Modeling cycle**
      - **Implementation cycle**

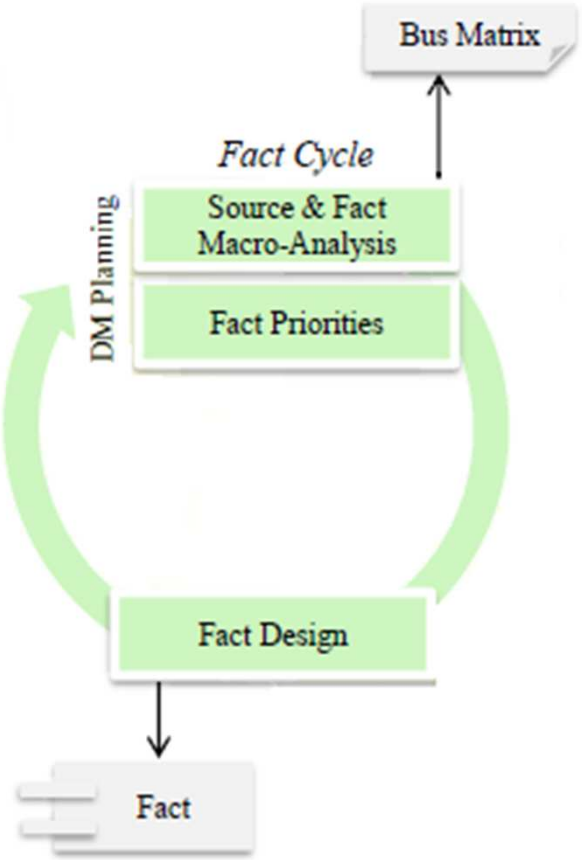


4WD: DM Cycle



- Architectural Sketch: overall functional and physical architecture of the data warehouse
- Conformity Analysis: definition of conformed dimensions across different facts and data marts
- DM Priorities: trade-off between user priorities and technical constraints
- DM Design: builds and releases the top-priority data mart

4WD: Fact Cycle



- Source & Fact macro-analysis: checks the availability, quality and completeness of the data sources and determines the main business facts
- Fact prioritization: trade-off between user requirements and technical priorities
- Fact design: develops and releases the top-priority fact

4WD principle applied:
1. Incrementality and Risk-Based Iteration

- Slicing the system functionality into **increments** (e.g. 2-4 weeks for a single fact release)
- **Risk guides** the data mart and fact priority definition

DM strategies

- Give priority to DM with widely shared hierarchies
- Prefer DM that are fed up from stable and well-understood data sources

Fact strategies

- Give priority to fact with the main business hierarchies and require the most complex ETL procedures
- Adopt data-driven approach
- Plan the length of an iteration in proportion to the complexity of the fact

4WD principle applied:
2. Prototyping

Use prototyping to support every data warehouse development phase:

- To help designers to **validate requirements**
- To improve the **design of reports** and **analysis applications**
- To **advance testing** to the early phases of design
- To **evaluate the feasibility** of alternative solutions during logical design of multidimensional schemata and during ETL design

4WD principle applied:
3. User involvement

Tight collaboration between users and designers:

- Preliminary **user training** (e.g. clarify project goals, explain the multidimensional model)
- Prototyping to favor **user awareness**
- **User feedback** to detect problems and errors
  - For usability tests of reporting and OLAP front-ends
  - For functional tests of ETL procedures

4WD principle applied:
4. Component reuse

Favor the use of predefined elements to support the data warehouse development:

- Conformed hierarchies
- Library hierarchies
- Library facts
- ETL building blocks
- Analysis templates

4WD principle applied:
5. Formal and light documentation

Formal but lean documentation to formalize requirements, simplify communication and support accurate design:

At the data warehouse level:

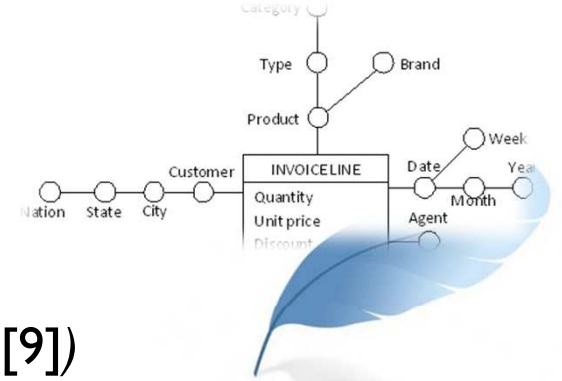
- **Effective schema** to summarize the data marts, data sources and user profiles

At the data mart level:

- **Bus matrix** to associate each fact with its dimensions

At the fact level:

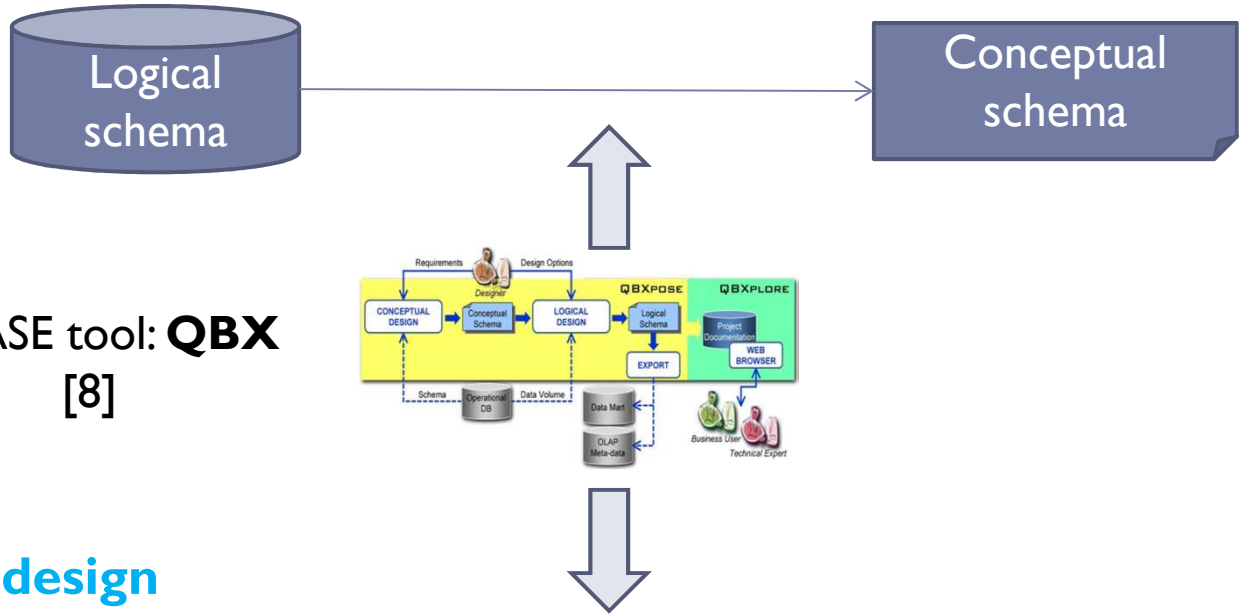
- **Conceptual schema** before proceeding with the implementation (e.g. *Dimensional Fact Model* (**DFM**) [9])



4WD principle applied:
6. Automated Schema Transformation

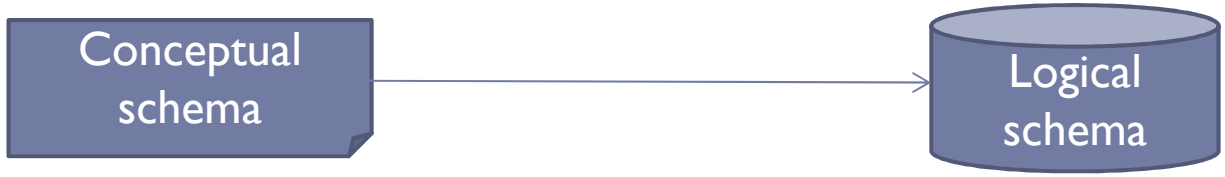
Automatic transformations between every data warehouse design level:

- **Supply-driven conceptual design**



A CASE tool: **QBX** [8]

- **Logical design**



Practical evidences

4WD was applied to a project in the area of pay-tvs

- 2 Data marts:
  - Administration: 9 facts, 5 releases
  - Management control: 3 facts, 2 releases
  - 10 to 26 days for each release

Benefit	Strategy
Project development speed-up	• User involvement • Prototyping
Reduction of the implementation effort	• Reusing of existing reports and dimension tables
Concise but exhaustive documentation	• DFM as conceptual model
Logical design automation	• CASE tool

Summary and Future work

- We have identified the **main problems** behind data warehouse projects and we have proposed an **innovative data warehouse methodology**
- We carry out a **case study** to assess the impact of 4WD in a real environment, but many practical extensions are possible:
  - Apply 4WD to different type of companies
  - Design a tool to support the analyst using the 4WD methodology

References

[1] Agile Manifesto: Manifesto for agile software development. http://agilemanifesto.org/ (2010)

[2] Boehm, B.W.: A spiral model of software development and enhancement. IEEE Computer 21(5), 61–72 (1988)

[3] Heineman, G.T., Councill, W.T.: Component-based software engineering: Putting the pieces together. Addison-Wesley (2001)

[4] Kruchten, P.: The 4+1 view model of architecture. IEEE Software 12(6), 42–50 (1995)

[5] Martin, J.: Rapid application development. MacMillan (1991)

[6] Pomberger, G., Bischofberger, W.R., Kolb, D., Pree, W., Schlemm, H.: Prototyping-oriented software development — concepts and tools. Structured Programming 12(1), 43–60 (1991)

[7] Royce, W.W.: Managing the development of large software systems: Concepts and techniques. In: Proc. ICSE. pp. 328–339. Monterey, California, USA (1987)

[8] Battaglia, A., Golfarelli, M. , Rizzi, S.. QBX: A CASE Tool for Data Mart Design. To appear on: Proceedings 30th International Conference on Conceptual Modeling (ER 2011) Brussels, Belgium, 2011.

[9] Golfarelli, M. , Rizzi, S.: *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill, 2009.

Thank you for your attention

Questions?