Database design

- Design goals
- From conceptual design to logical and physical design

Jouni Huotari

jamk.fi

Database design

Why is it so important?

•© Jouni Huotari & Ari Hovi

Why is it important to focus on the design of the database?

- The database(s) must serve the functions of the organization
- The database is an essential part of business-oriented systems
- The users see user interfaces, reports and other parts of the system (and thus data must be correct from right places)
- If the database behind is not well designed, the application itself will never be very good
- There will be extra work in the programming phase if the data structures are not well designed
- On the other hand, if the database is designed properly, the programming of the application is easier

© Jouni Huotari & Ari Hovi

Properties of a well-designed database

- Offers many ways to fetch data
- Flexible, works well and gives the right data
  - Consistency => see ACID
  - Integrity, see SQL (e.g., referential integrity, CHECK)
- Adapts itself to the data growth in the future
  - it should be the possible to change the structure of the database (more tables or more columns in a table) with minimum changes to existing applications
  - a new database should work in cooperation with the other systems and databases of the organization.

© Jouni Huotari & Ari Hovi

Design goals

There are many goals for the design of a database; here are some:

1. The database is comprehensive: it includes all the needed data and connections.
2. The database is understandable: there is a clear structure which leads to easy, flexible and fast reading and updating of the data.
3. The database is expandable: it is possible to change the structure of the database with a minimum change to the existing software.
4. The database can be used in many organizations: the database can be adapted to different kinds of environments and customers without the need to change the database structure.
5. The integrity of the data: data must be correct; it must be consistent.
6. The application design and implementation is easy and fast.

© Jouni Huotari & Ari Hovi

Characteristics of a well-designed data model

Data Model Scorecard by Steve Hoberman:

- How well does the model capture the requirements?
- How complete is the model?
- How well does the model match its scheme?
- How structurally sound is the model?
- How well does the model leverage generic structures?
- How well does the model follow naming standards?
- How well has the model been arranged for readability?
- How good are the definitions?
- How consistent is the model with the enterprise?
- How well does the metadata match the data?

The five killer questions (by Steve Hoberman)

1. Is the scope of the model clear?
2. Is there a well-defined purpose as to why the model was built?
3. Has the correct filter been applied?
4. Does the model show the correct time view?
5. Is there any functionality in the requirements that is not present on the model?

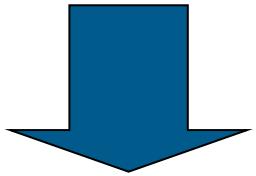
From conceptual design to logical and physical design

jamk.fi

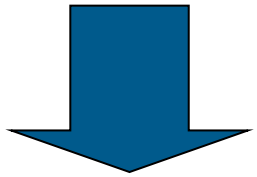
Three phases of database design

Conceptual design => CDM
(Conceptual Data Model)

Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

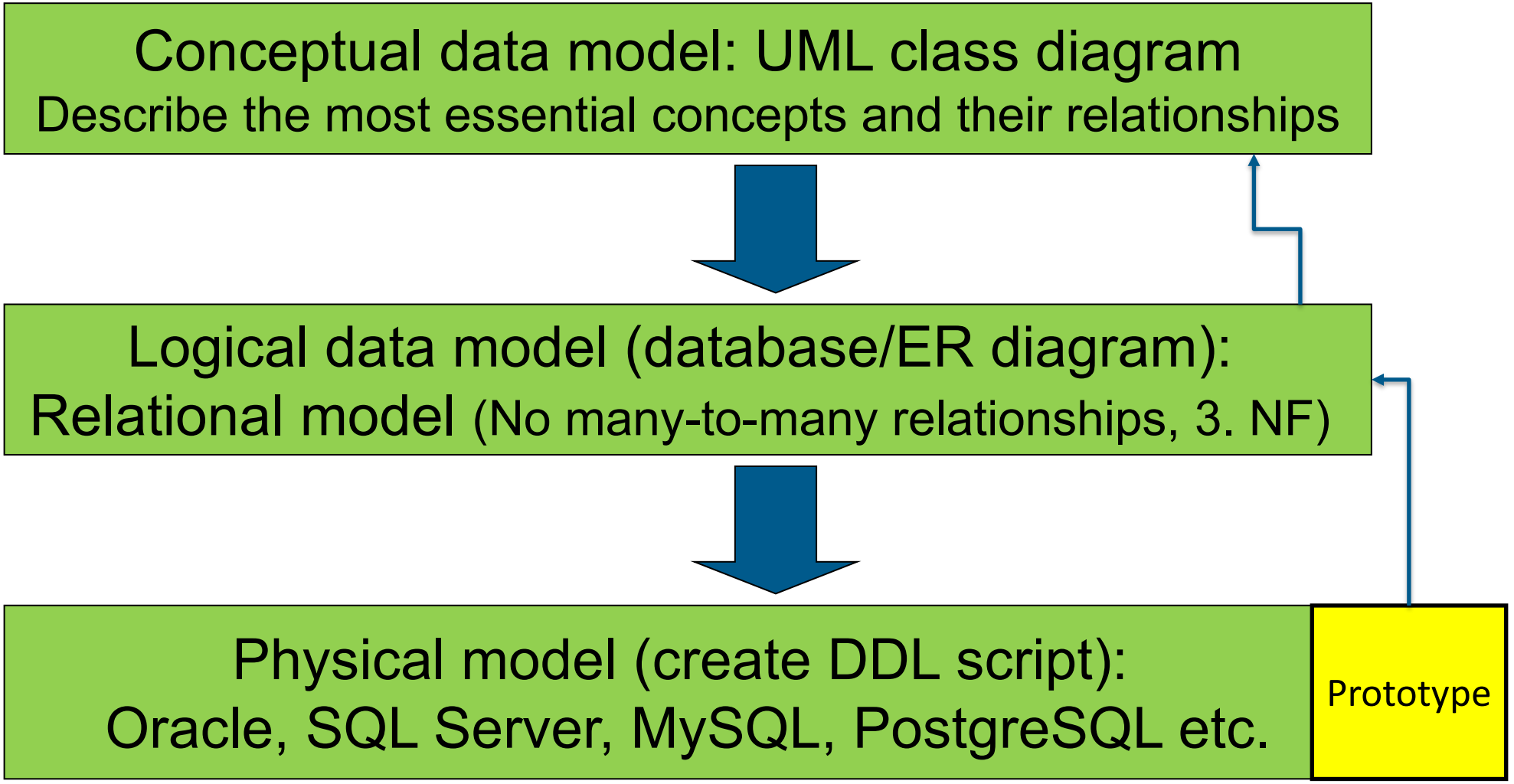


Logical design: choose a data model
(LDM, Logical Data Model)



Physical design: select the DBMS
(PDM, Physical Data Model)

From conceptual data model to physical data model



Data models defined

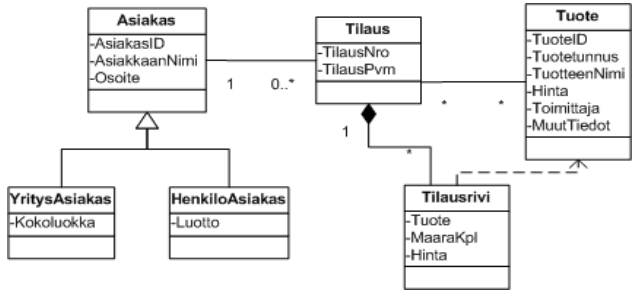
Conceptual Data Model (CDM)	Logical Data Model (LDM)	Physical Data Model (PDM)
Includes high-level data constructs	Includes entities (tables), attributes (columns/fields) and relationships (keys)	Includes tables, columns, keys, data types, validation rules, database triggers, stored procedures, domains, and access constraints
Non-technical names, so that executives and managers at all levels can understand the data basis of Architectural Description	Uses business names for entities & attributes	Uses more defined and less generic specific names for tables and columns, such as abbreviated column names, limited by the database management system (DBMS) and any company defined standards
Uses general high-level data constructs from which Architectural Descriptions are created in non-technical terms	Is independent of technology (platform, DBMS)	Includes primary keys and indices for fast data access.

For more information: DoDAF DM2

jamk.fi

From CDM (class diagram) to tables (PDM)

CDM as Class diagram



LDM as ER diagram (crow's feet)

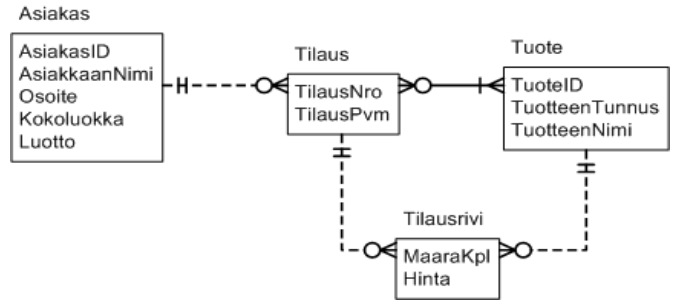
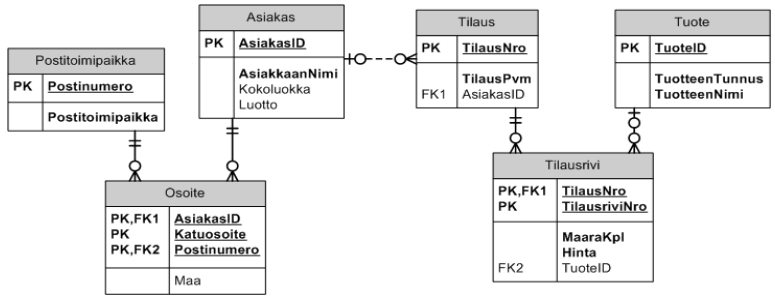
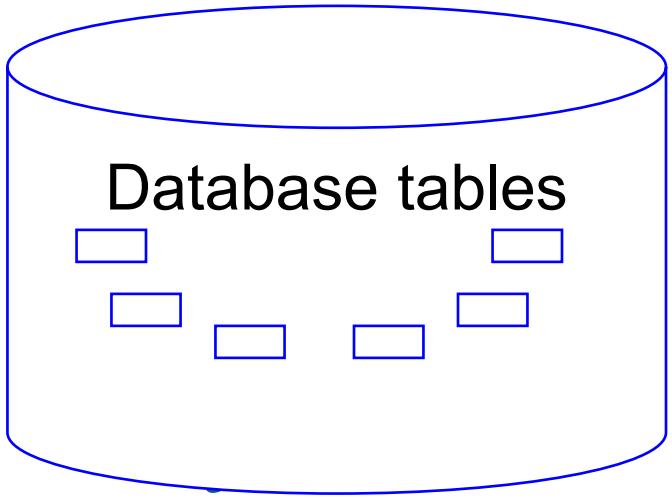


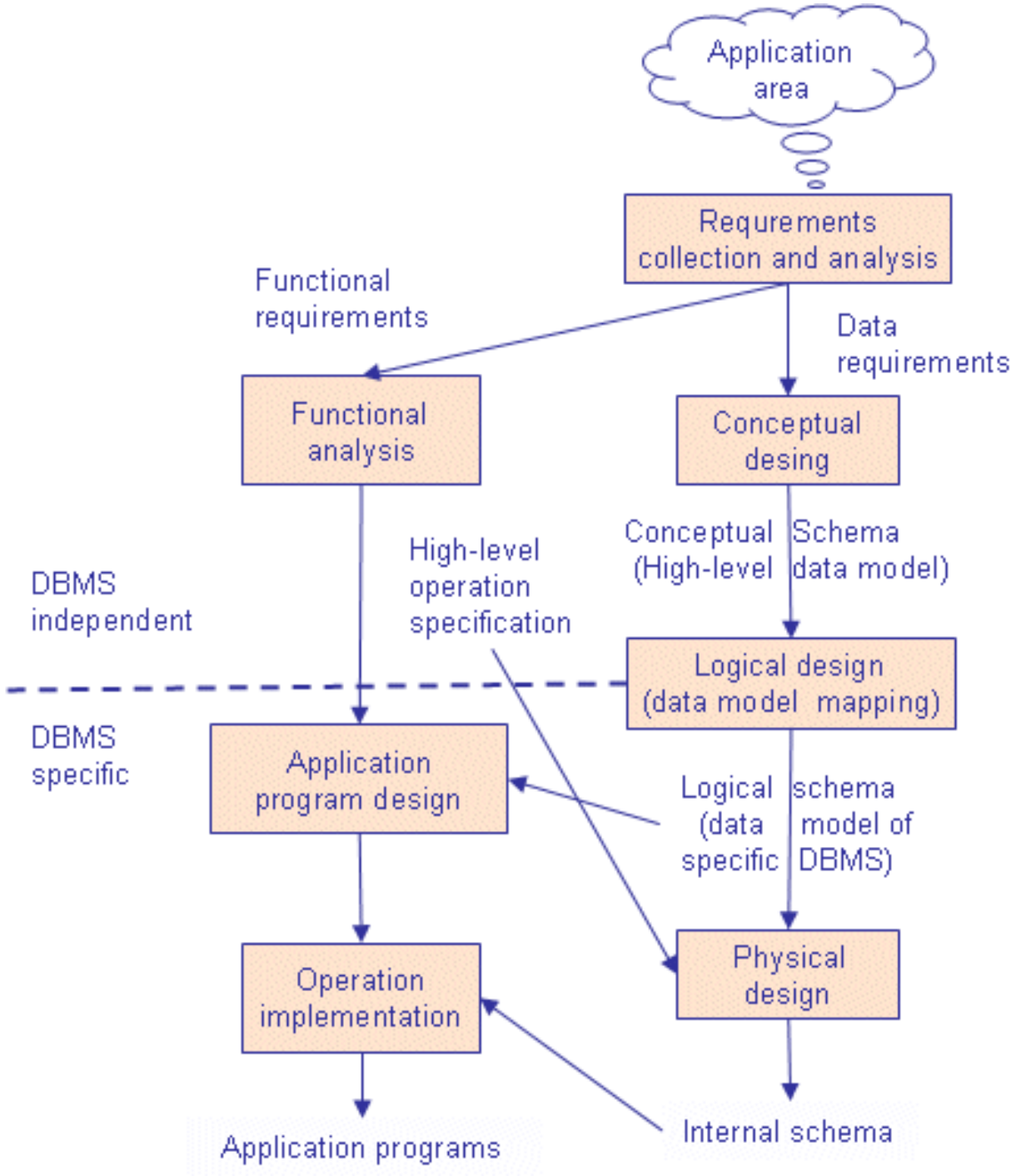
Table schema (3 NF)



Database tables



© Jouni Huotari & Ari Hovi


Application area
Requrements collection and analysis
Functional requirements
Data requirements
Functional analysis
Conceptual desing
Conceptual (High-level
Schema data model)
High-level operation specification
DBMS independent
Logical design (data model mapping)
DBMS specific
Application program design
Logical (data specific
schema model of DBMS)
Operation implementation
Physical design
Application programs
Internal schema

CDM

LDM

PDM