

Database Modeling

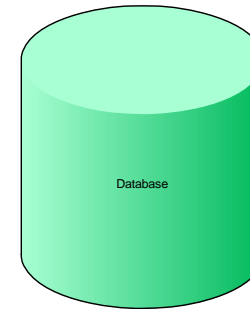
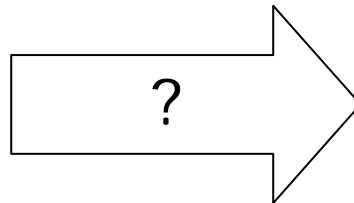
DBTech-EXT - Thessaloniki, Greece 10.9.2009

Outi Virkki - Haaga-Helia University of Applied Sciences

Contents

How to accomplish a database ?	2
Toolbox for a database designer	3
Methodology !	3
Other Tools ?	3
Software Lifecycle / Systems Development Process.....	4
Database Design Methodology / Database Design Process.....	5
Differences between Conceptual and Logical Schema	6
Multiple notations.....	7
Example case: Experts	8
Conceptual Modeling	9
Conceptual Schema	9
Repository / Data dictionary.....	9
Logical Modeling.....	10
Relational Schema.....	10
Integrity Constraints.....	10
Normalization ?.....	10
Physical Modeling.....	11
Security.....	11
Performance.....	11
Summary & Quality considerations	12
Reverse Engineering & Re-engineering	13
CASE tools	14

How to accomplish a database ?



```
CREATE TABLE table1  
(id_col integer not null,  
col1 varchar(20) not null,
```

```
...  
);
```

```
CREATE TABLE table2  
(...);
```

Toolbox for a database designer

Methodology !



Your main tool is to **know what to do !**

- ← **3-phase methodology** is recommended by all reckoned database books - and professionals

Other Tools ?

- ← to support the methodology / design process
- ← to document (requirements, ideas, specifications, ...)
- ← to draw diagrams
- ← to communicate the design visually (diagrams etc.)
- ← to generate code ?
- ← to ease your work !

CASE – Computer Aided Systems Engineering provides assistance

- ~ set of tools and methods to assist Systems Development

Software Lifecycle / Systems Development Process

1. Requirements Definition / Analysis / Engineering
↓
2. **System Design**
↓
3. Implementation / Construction
↓
4. Testing
↓
5. Deployment
↓
6. Maintenance



← Where does database design fit in ?

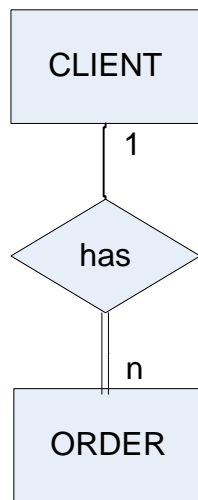
Database Design Methodology / Database Design Process

Level	Objective	Main Tasks	Outcome
Conceptual Modeling	<p>To understand the behavior and concepts of the target system</p> <p>Main concepts, their meaning and relationships</p>	Conceptual Schema Design	Conceptual Schema / Class Diagram Repository
↓			
Logical Modeling	<p>To present the gained knowledge by means of the chosen data model (relational model)</p>	Logical Schema Design Normalization	Logical Schema / Relations Integrity Constraints
↓			
Physical Modeling	To provide appropriate storage structures and security mechanisms	Security Design Performance Design	SQL-statements to create tables, constraints, indexes, sequences, storage parameters, privileges, roles, views, triggers, ...

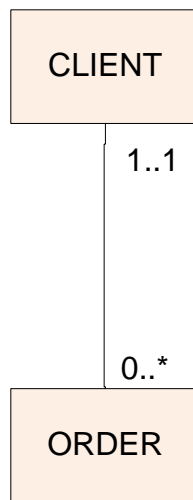
Differences between Conceptual and Logical Schema

Conceptual schema	Logical shema = Relational schema
To understand and specify the objects, structure and activity of the target system	To specify the objects of the target system according to relational model
Specifies the object classes / entity types and relationships between them	Specifies relations (and attributes and references) according to relational model
Primary key for each class / entity type	Primary key for each relation
No foreign keys needed (but allowed)	Foreign keys implement the references between relations
Class hierarchy and aggregation OK	only associations allowed
many – to - many – relationships allowed	many – to - many – relationships not allowed
data types need not be specified	data types specified

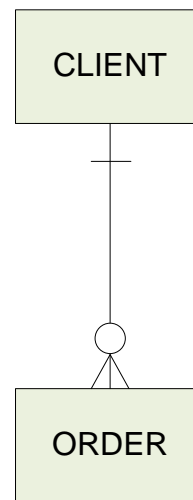
Multiple notations



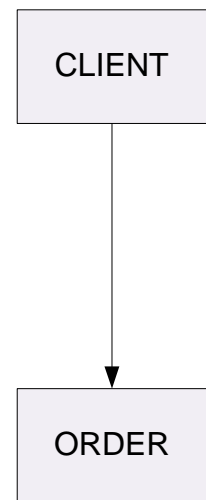
ER



UML



Crow's feet



Arrow

← same contents (to great extent)

Example case: Experts

There is a group of Database professionals who would like to share their expertise with each other and learn from each other.

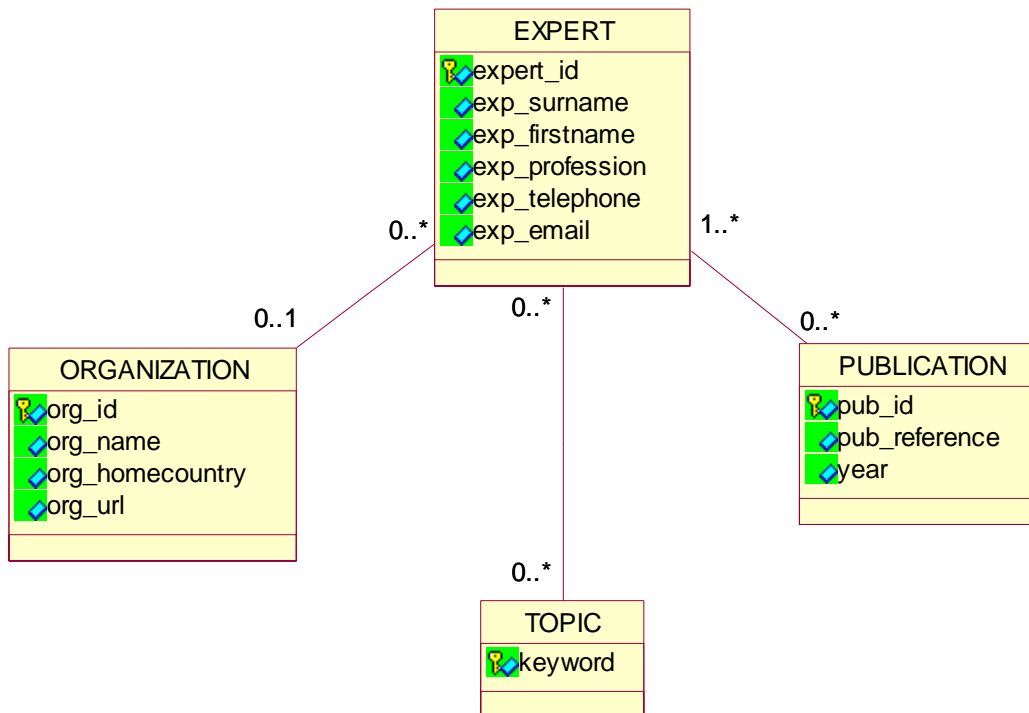
They want to find easily

- the contact information of an expert
- fields of interest of an expert
- publications of an expert



Conceptual Modeling

Conceptual Schema

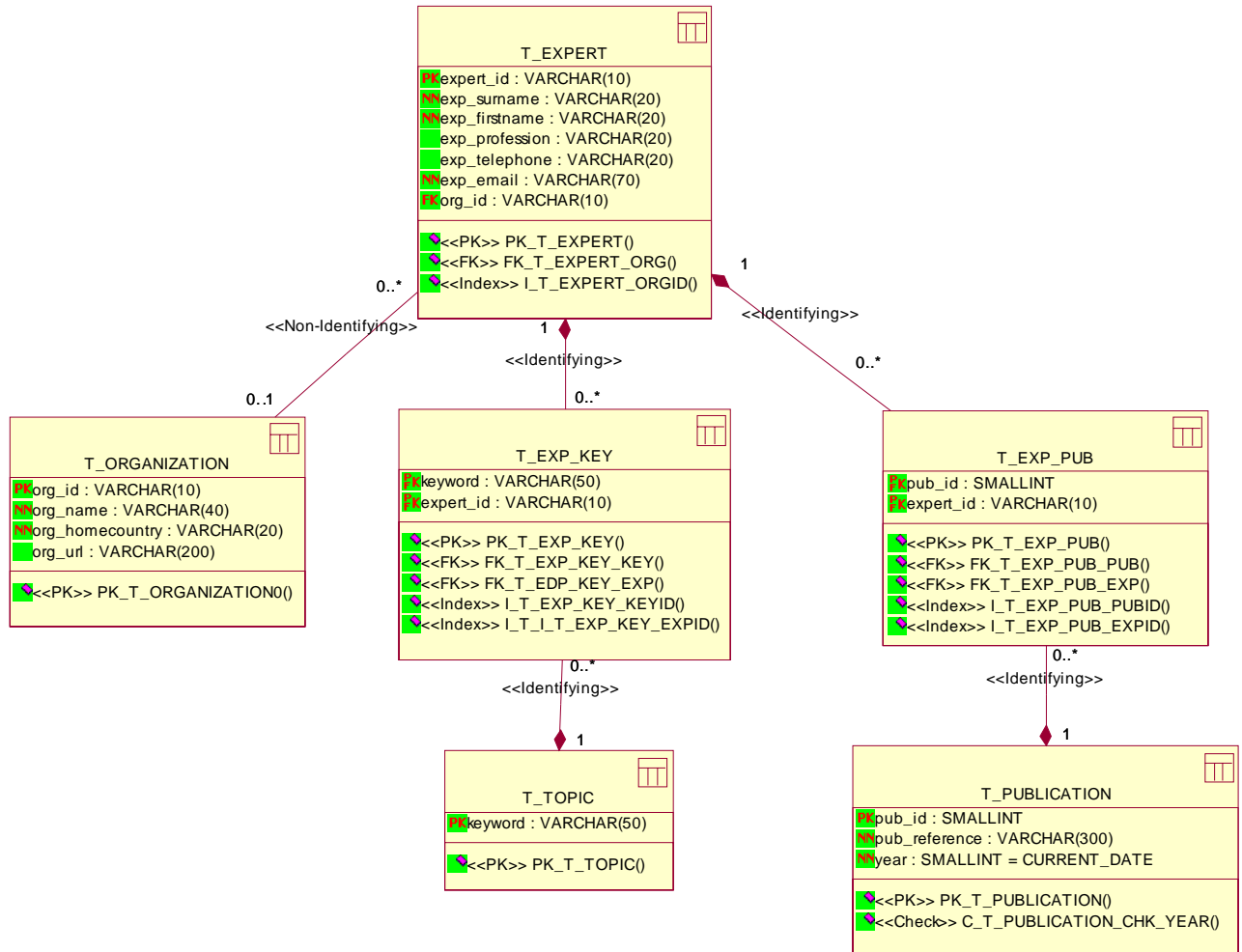


Repository / Data dictionary

Class / Entity Name	Expert
Definition	a person with extensive knowledge or skill in a particular area of interest
Identified by	expert_id
Attributes:	exp_surname exp_firstname exp_profession exp_telephone exp_email exp_memo

Logical Modeling

Relational Schema



Integrity Constraints

- ✓ Data types, Primary Keys, Foreign Keys
- ✓ Check constraints
- Triggers ? Assertions ?

Normalization ?

1.-5. Normal Forms to check the design for anomalies

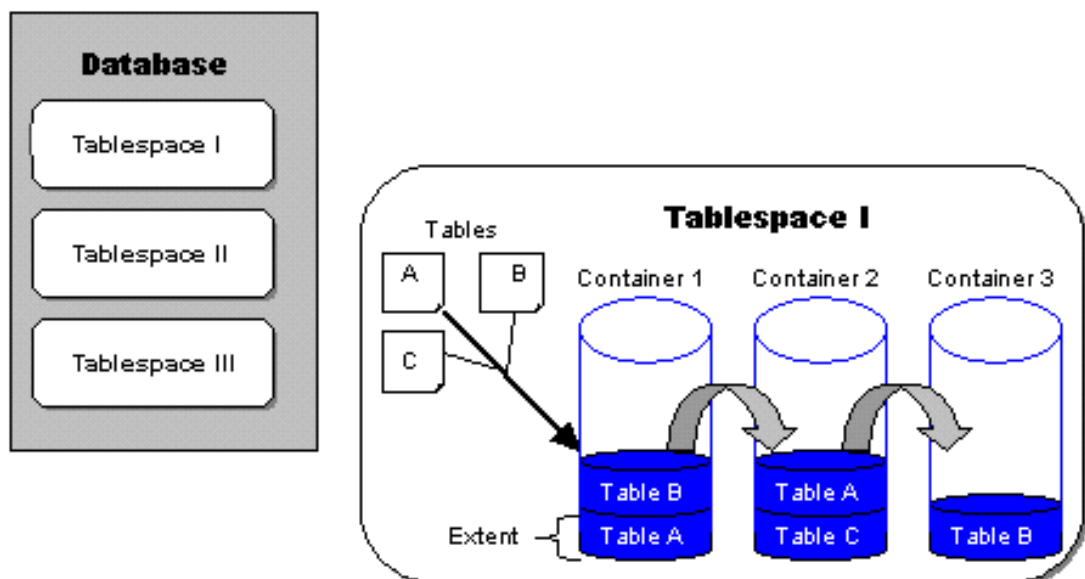
Physical Modeling

Security

Security mechanism	Rational Rose Data Modeler support
views	OK
roles	~NO
priviledges	~NO

Performance

Performance booster	Rational Rose Data Modeler support
indexes	OK
sequences for primary keys	~NO
storage allocation	creating tablespaces OK assigning tables to tablespaces OK



Summary & Quality considerations

Quality / User

Meets user's needs

Reliability

Efficiency

Quality / Developer

Maintainability

Compatibility

Portability

Reusability

Localizability

Expandability



Reverse Engineering & Re-engineering

- A database without documentation ?
- A database with outdated documentation ?

CASE-tool may help - - to

Reverse Engineer

- ← create representations of the system at a higher level of abstraction
- ← visualize a complex / large database
- ← go backwards the development cycle



Re-engineer

- make modifications to the design
- produce a new implementation



CASE tools

Enforce standards (methodology, naming, ...) !

Increase productivity (diagrams, automation, ...) !

Centralized repository for specifications !

Products

- IBM Rational Rose
- Star UML
- IBM Infosphere Data Architect
- Oracle SQL Developer Data Modeler
- ER Win
- DeZign
- ...

Classification	
Upper CASE Tools	<ul style="list-style-type: none">• support strategic / higher level software engineering• support notations like ER diagrams, UML , Data flow diagrams, etc.
Lower CASE Tools	<ul style="list-style-type: none">• concentrate on the back end activities of the software engineering• support physical design, debugging, construction, testing, integration of software components, maintenance, reengineering and reverse engineering activities.
Integrated CASE Tools	<ul style="list-style-type: none">• support the whole software engineering and provide the functionalities of both upper and lower CASE tools