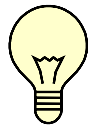
Measures for Schema Quality

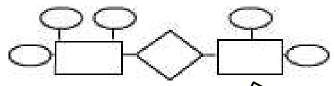
Hans J. Lenz, Freie Universität Berlin

September 2007

1 Workflow: From Kick-off to DB System

Demand for mapping reality into a „mini world“.

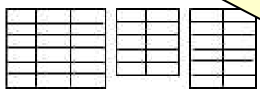




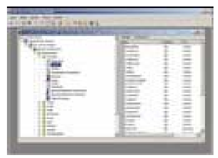
Conceptual Design und Visualisation
(conceptual Data Model)

Schema Quality

Normalised Relational Model
(logical Data Model)



Implementing DB System



2 Illustration: Modelling Variants

Person

ID	Name	Surname	Address
1	John	Smith	113 Sunset Avenue 60601 Chicago
2	Mark	Bauer	113 Sunset Avenue 60601 Chicago
3	Ann	Swenson	4 Heroes Street Denver

(a)

Person

ID	Name	Surname
1	John	Smith
2	Mark	Bauer
3	Ann	Swenson

Address

ID	StreetPrefix	StreetName	Number	City
A11	Avenue	Sunset	113	Chicago
A12	Street	4 Heroes	null	Denver

ResidenceAddress

PersonID	AddressID
1	A11
2	A11
3	A12

(b)

Source: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

2 Illustration: Problems

Problem of Modelling (a):

- **ambiguous values**
- **redundant data**
- **superfluous info**

trade-off

Problem of Modelling (b):

- **complex Structure**

3 Seven Dimensions of Schema Quality

1. Readability
2. Normalisation
3. Correctness w.r.t. Model
4. Correctness w.r.t. Requirements
5. Minimalisation
6. Completeness
7. Pertinence („over modelling“)

Source: Redman (1996)

3.1 Readability of ERM / UML

DEF.: A schema is readable whenever it represents the meaning in the reality represented by the schema in a clear way for its intended use.

Aesthetic Criteria

- **Avoid Crossing between arcs (prefer planar graph)**
- **Embed symbols in a grid**
- **Horizontal or vertical drawings of lines mandatory**
- **Minimum number of bends of lines**
- **Minimum Area of Diagram (one glimpse capturing)**

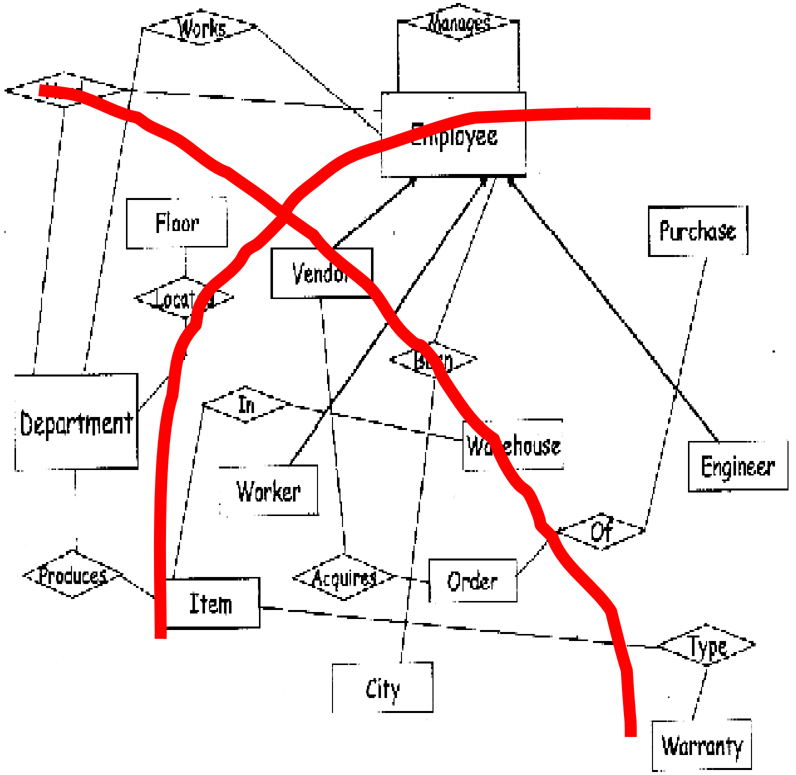
Structural Adequacy

- **Hierarchical Representations of Objects**
- **Symmetry of Children-Objects w.r.t. Parent-Objects**

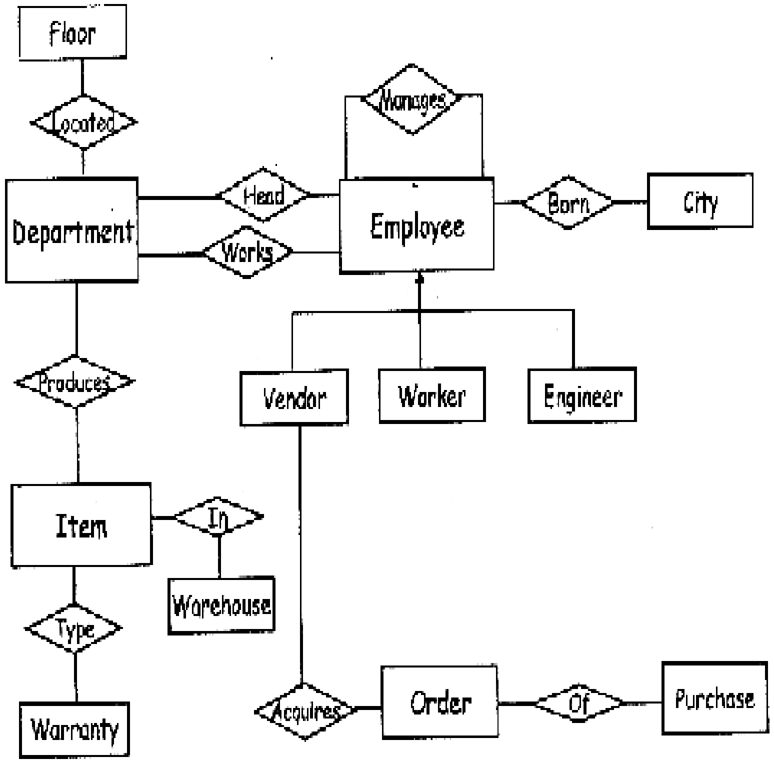
Source: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3.1 Readability of ERM

„Spaghetti"-Style:

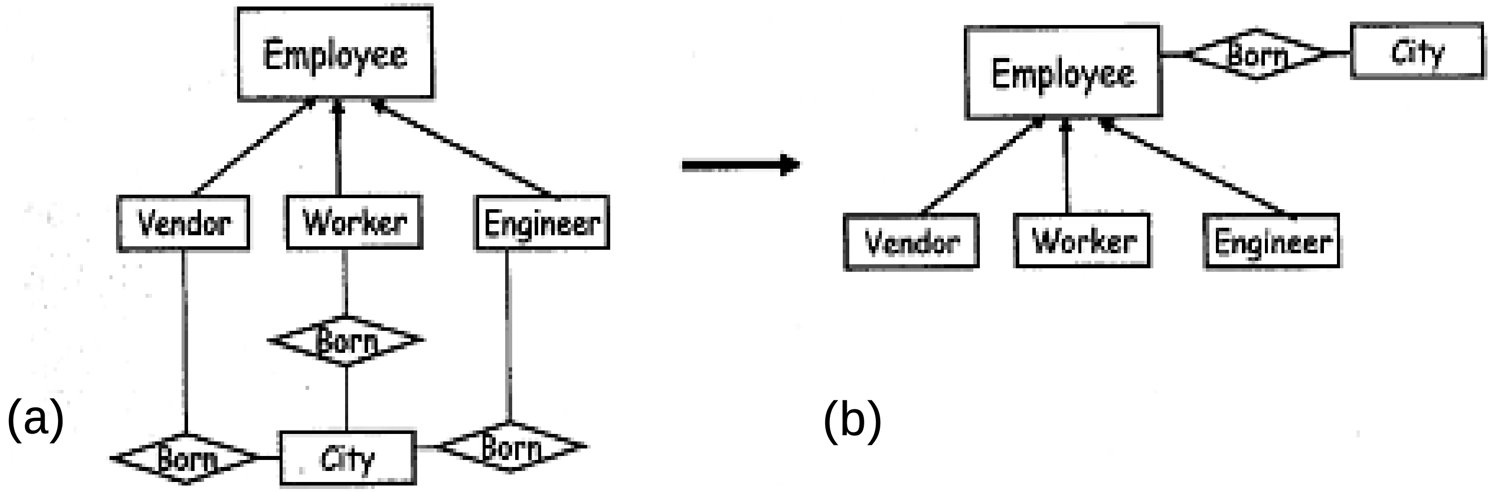


Equivalent readable Schema:



Quelle: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3.1 Readability of ERM (2)



Two equivalent models showing is-a generalisation.

Compactness of (b) due to inheritance.

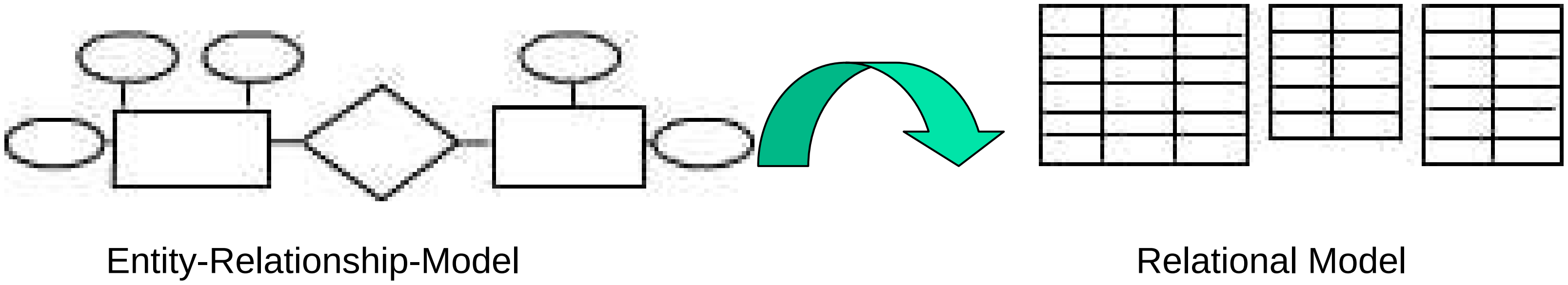
Quelle: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3 Seven Dimensions of Schema Quality

1. Readability ✓
2. Normalisation
3. Correctness w.r.t. Model
4. Correctness w.r.t. Requirements
5. Minimalisation
6. Completeness
7. Pertinence (over modelling)

3.2 Normalisation

DEF.: Loss-less Decomposition of a relational model (set of tables) in order to avoid redundancy and anomalies of data management



Modelling:

- **intuitive / rules of thumb / logical criteria**
- **identification of structural weakness**
- ***informal* (heuristic) or *formal* (Normalisation) criteria of correct relational designs**

Ex.: Structural deficits of a schema:

Employee

M-Nr	M-Name	M-GebDat	A-Nr	A-Bez	A-Leiter
234	Müller	1.10.1959	1	Einkauf	~~234~~ 376
~~345~~	~~Meier~~	~~30.3.1961~~	~~2~~	~~Marketing~~	~~245~~
376	Schmidt	15.6.1968	1	Einkauf	234 !
~~245~~	~~Schulz~~	~~31.5.1965~~	~~2~~	~~Marketing~~	~~245~~
<NULL>	<NULL>	<NULL>	3	Produktion	<NULL>

Update-Anomaly:

Inconsistencies if changes are not effective across full database .

insert-Anomaly (Entity Integrity Constraint):

Null values not allowed!

Delete-Anomaly:

Loss of Information about facts

3.2.2 Formal Criteria of Modelling

Normalisation

Steps of sequenced decomposition of relation types into subtypes

→

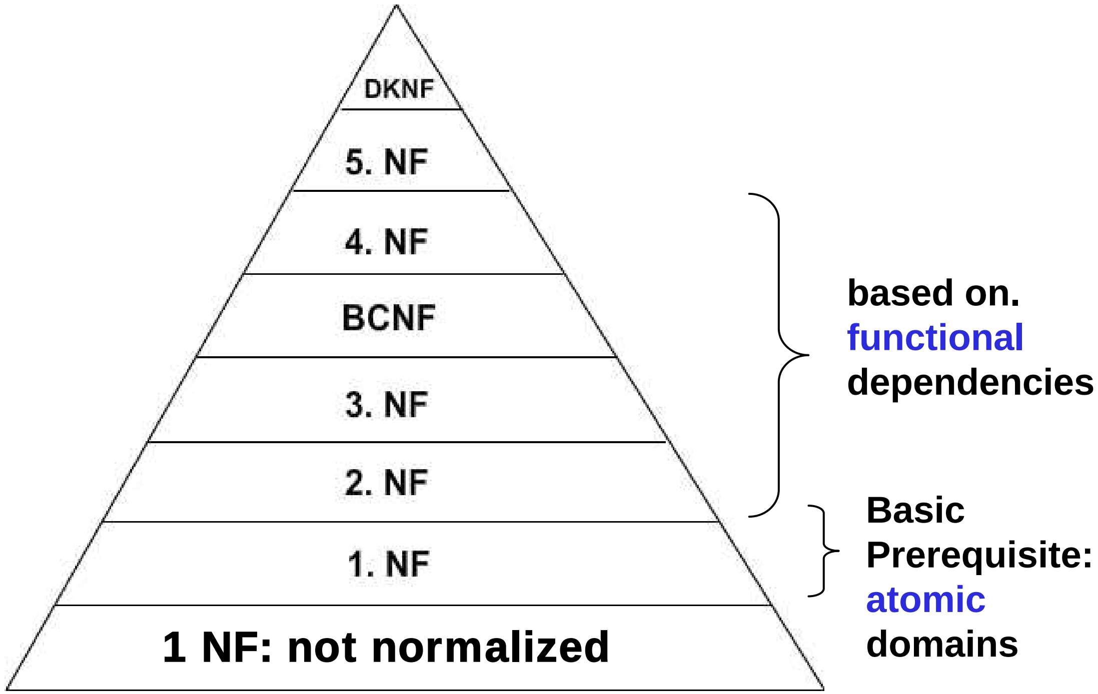
Normal Form

State of a Relation type
Represents quality of design

3.2.3 Hierarchy of Normal Forms

Normal Forms
are stacked

..Relations of level k
satisfy restrictions of
level h < k = 1, 2,...,5



3.2.3 Normal Forms

- **DEF.: 1. Normal Form (1NF)**
  All Attribute values of a schema must have atomic data types, i.e. sets, bags, arrays, records, lists, tables etc. not allowed

Ex.: 0-NF

Person

Nr	Name	Adresse
234	Müller, Hans	Bismarkstr. 11, 10961 Berlin
345	Meier, Otto	Hüttenweg 32, 10944 Berlin
376	Schmidt, Jan	Bergmannstr. 25, 10174 Berlin

... Attribute *Name* is a concatenated string of family and first name

… Attribute *Adress* is a record

3.2.3 Normal Forms (2)

DEF.: Normalisation

map a set-valued attribute into a set of single-valued attributes

Poor Quality Solution:

use single attributes for each item. Note that the group assignment is lost

Good Quality Solution:

Define a separate table schema and link it to the original table by a foreign key - primary key relationship.

3.2.3 Normal Forms (3)

DEF.: Functional Dependency (FD)

Attribute B is functional dependent on attribute A, if for each value of A there exists only a unique value of B (true for groups *of* attributes, too).

Ex.:

Teamwork

<u>Emp#</u>	Emp-Name	<u>Pro#</u>	Pro-Name	Work_ Load

Partial FD

Partial FD

Full FD

3.2.3 Normal Forms (4)

- **DEF.: 2nd Normal Form (2NF)**
  Table Schema is in 1NF and each non-key attribute must be fully dependent on each candidate key.

1NF:

Teamwork

MA#	MA-Name	Pro#	Pro-Name	Stunden

1NF but not 2NF

2NF:

Teamwork

MA#	Pro#	Stunden

Employee

MA#	MA-Name

Project

Pro#	Pro-Name

1NF and 2NF

3.2.3 Normal Forms (5)

DEF.: Transitive Dependency

Attribute C is transitive dependent on candidate key A, if a non-key attribute B exists on which C is functional dependent where B itself is functional dependent on A.

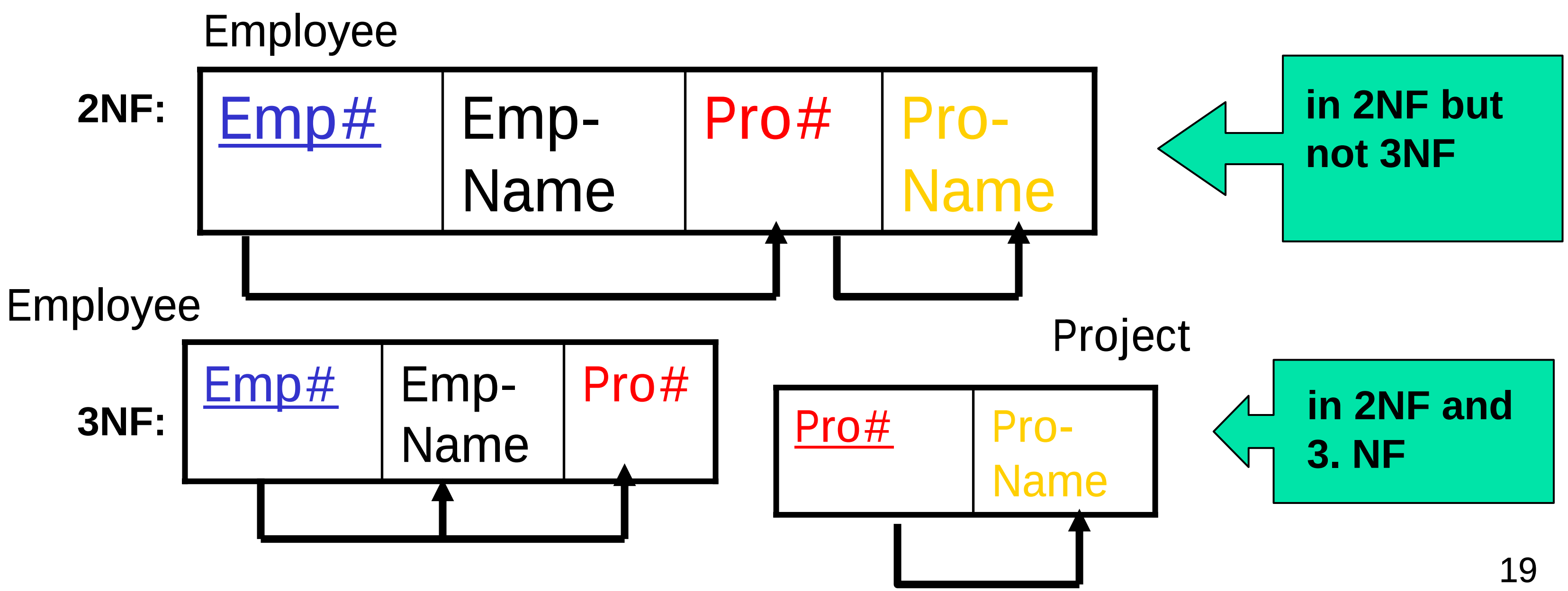
Ex.: Employee

<u>Emp#</u>	Emp-Name	Pro#	Pro-Name

3.2.3 Normal Forms (6)

- **DEF.: 3rd Normal Form (3NF)**

  Table schema is 2NF and no non-key attribute is transitive dependent on any candidate key.



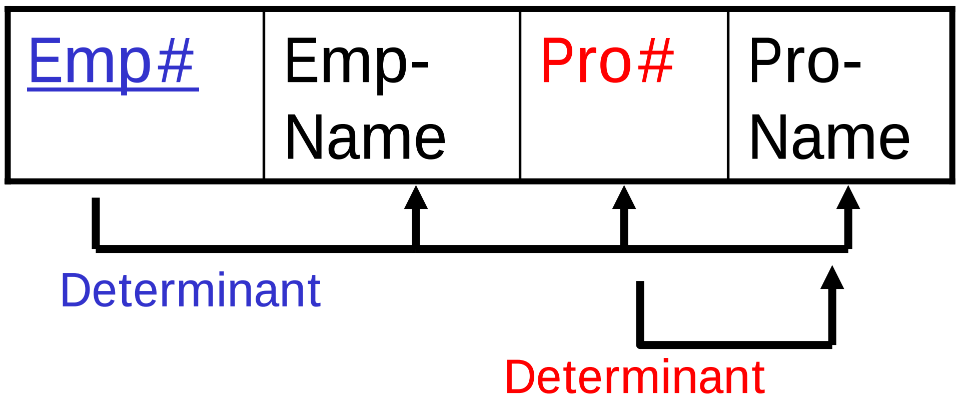
3.2.3 Normal Forms (7)

DEF.: Determinant

Attribute A is a determinant if there exists at least another attribute B which is fully dependent of A.

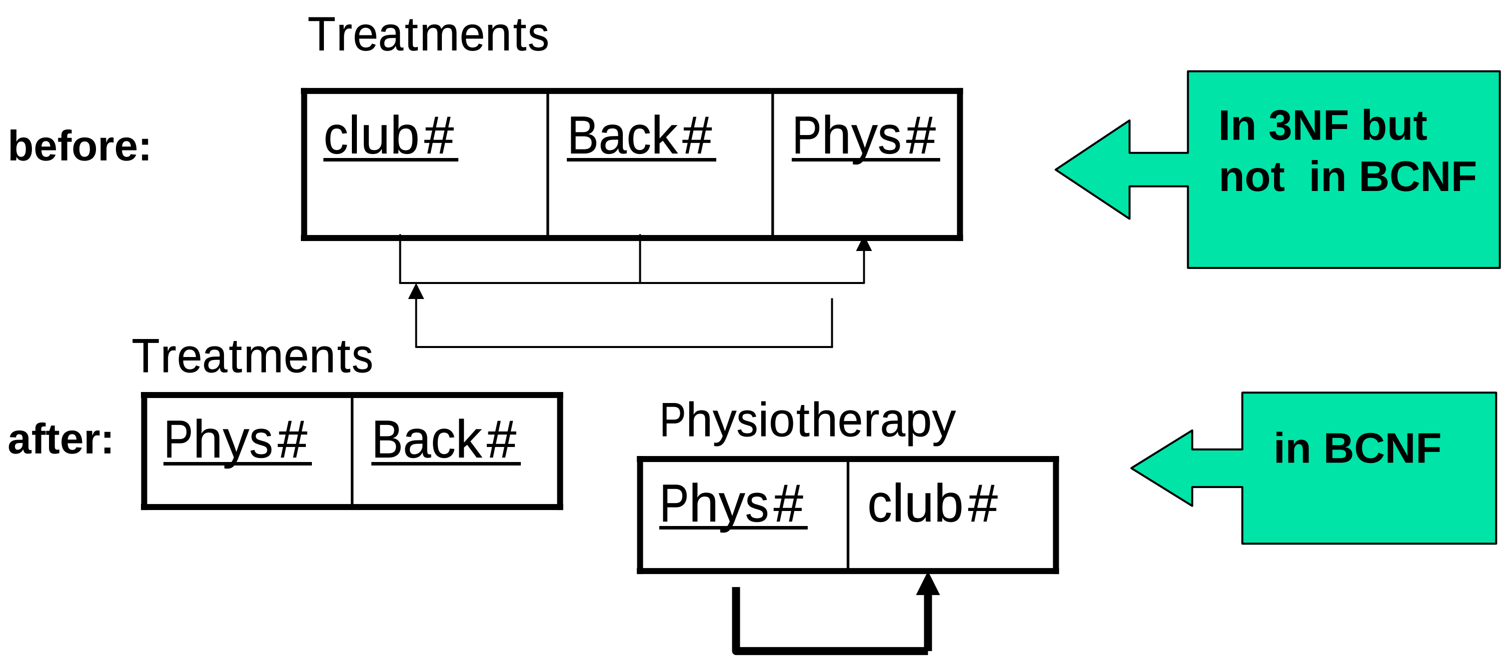
Employee

Ex.:



3.2.3 Normal Forms (8)

- **Boyce-Codd-Normalform (BCNF)**
  A table schema is in Boyce-Codd Normal Form if each determinant is a candidate key.

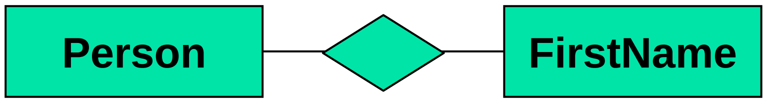


3 Seven Dimensions of Schema-Qualität

1. **Readability** ✓
2. **Normalisation** ✓
3. **Correctness w.r.t. model**
4. **Correctness w.r.t. requirements**
5. **Minimalisation**
6. **Completeness**
7. **Pertinence (over modelling)**

3.3 Correctness w.r.t. Model

- **DEF.: Correct Modelling as far as requirements are concerned**



Not correct:
Object *FirstName* is not a real object but a category.

Person:

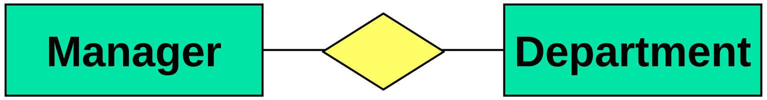
Per#	First Name	Family Name
1	Klaus	Meier
2	Hans	Müller
3	Otto	Schmidt

Correct:
***FirstName* as Attribute in schema *Person*“.**

Quelle: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3.4 Correctness w.r.t. Requirements

- **DEF.: Corretness w.r.t. to requirements is the correct representation of constraints / requirements in terms of object categories**



1 : 1 OK

1 : n wrong!

Business Rule:
Each department is headed by exactly one manager and each manager is the head of exactly one department.

Quelle: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3.3-3.4: Mis-Specification of a cube
The Cattle-Example of Snodgrass

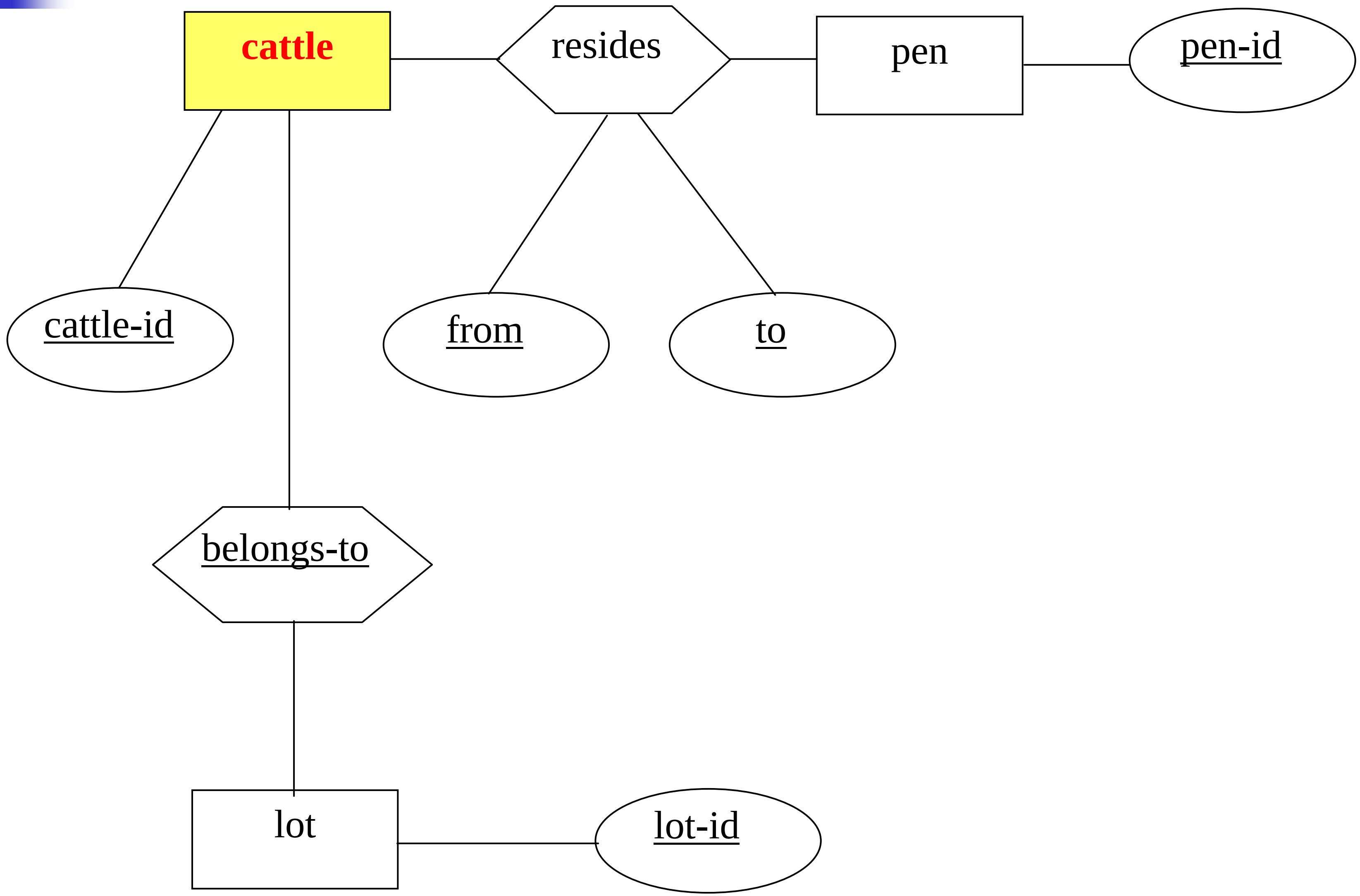
- Snodgrass (1999) defines a data cube (4-way contingency table) "Count of cattle grouped by lot, pen and date“
- The categorical attribute (dimension) '*dat*e' is split into the two sub-attributes *'from_date'* and *‘to_date’*.
- Fact Table:

count
<u>lot-id</u>
<u>pen-id</u>
<u>from-data</u>
<u>to-data</u>

Source: Sno99, chap. 11 "Conceptual Design".

ERM of "Cattle"-Example

(revised version on OLTP level)



Source: Lenz and Thalheim (2002)

Relational Modelling

Model by Snodgrass (1999)

FDYD (Fdyd_ID, Name,...)

LOT(Fdyd_ID, Lot_ID-Num, Lot_Id, Gndr_Code,...)

Pen(Fdyd_ID, Pen_ID,Pen_Type_Code,...)

Application (A_Name, A_Description,...)

DBF_File(A_Name, DBF_Name,...)

BKP(Fdyd_ID, BKP_Id, ...)

Model by Lenz and Thalheim (2002)

Cattle (Cattle_ID, BelongsTo, ...)

Lot (Lot_ID, ...)

Resides (Cattle_ID, Pen_ID, From, To, ...)

Pen (Pen_ID, ...)

Query: "Find the History of Lots being co-resident in a Pen"

```
select L1.Lot_Id_num, L2.Lot_Id_Num, L1.Pen_Id, L1.From_Date, L1.To_Date
          from Lot_Loc as L1, Lot_Loc as L2
          where L1.Lot_Id_num< L2.Lot_Id_num
             and L1.Fdyd_Id = L2.Fdyd_Id and L1.Pen_Id= L2.Pen_Id
             and L1.From_Date= L2.From_Date and L1.To_Date<= L2.To_Date
 union
select L1.Lot_Id_num, L2.Lot_Id_Num, L1.Pen_Id, L1.From_Date, L2.To_Date
          from Lot_Loc as L1, Lot_Loc as L2
          where L1.Lot_Id_num< L2.Lot_Id_num
                    and L1.Fdyd_Id = L2.Fdyd_Id and ...
union
select L1.Lot_Id_num, L2.Lot_Id_Num, L1.Pen_Id, L2.From_Date, L1.To_Date
          from Lot_Loc as L1, Lot_Loc as L2
          where L1.Lot_Id_num< L2.Lot_Id_num
                    and L1.Fdyd_Id = L2.Fdyd_Id and ...
union
select L1.Lot_Id_num, L2.Lot_Id_Num, L1.Pen_Id, L2.From_Date, L2.To_Date
          from Lot_Loc as L1, Lot_Loc as L2
          where L1.Lot_Id_num< L2.Lot_Id_num
                    and L1.Fdyd_Id = L2.Fdyd_Id and L1.Pen_Id = L2.Pen_Id
                    and L1.From_Date > L1.From_Date and L2.To_Date <= L1.To_Date;
```

Query based on Snodgrass mispecified Model

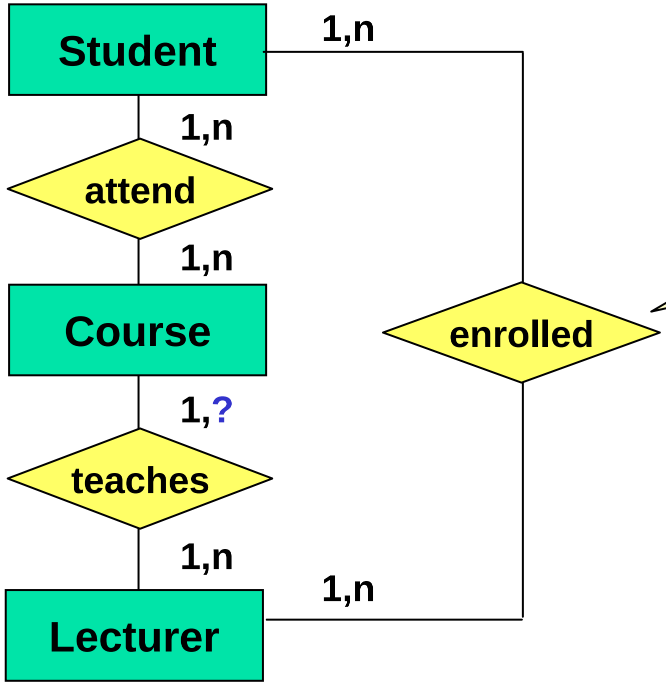
Query: "Find the History of Lots being co-resident in a Pen"

Query based on Lenz and Thalheim (2002) Model

```
select distinct L1.Lot_ID, L2.Lot_ID, R1.Pen_ID, R2.From, min(R1.To,
R2.To)
  from Cattle C1, Cattle C2, Resides R1, Resides R2, Lot L1, Lot L2
  where L1.Lot_ID = C1.BelongsTo  and  L2.Lot_ID = C2.BelongsTo   and
     R1.Cattle_ID = C1.Cattle_ID  and
     R2.Cattle_ID = C2.Cattle_ID  and  R1.Pen_ID = R2.Pen_ID  and
     R1.From <= R2.From and R2.From < R1.To and L1.Lot_ID <>
L2.Lot_ID.
```

3.5 Minimalisation

- **DEF.: A Schema is minimal if each part of the requirements is represented only once.**



Relational Type *enrolled* is redundant iff upper bound of cardinality „?“ = 1.

Quelle: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3.6 Completeness

- **DEF.: Extent to which a schema includes all objects necessary to meet some specified conceptual requirements**

Ex.:

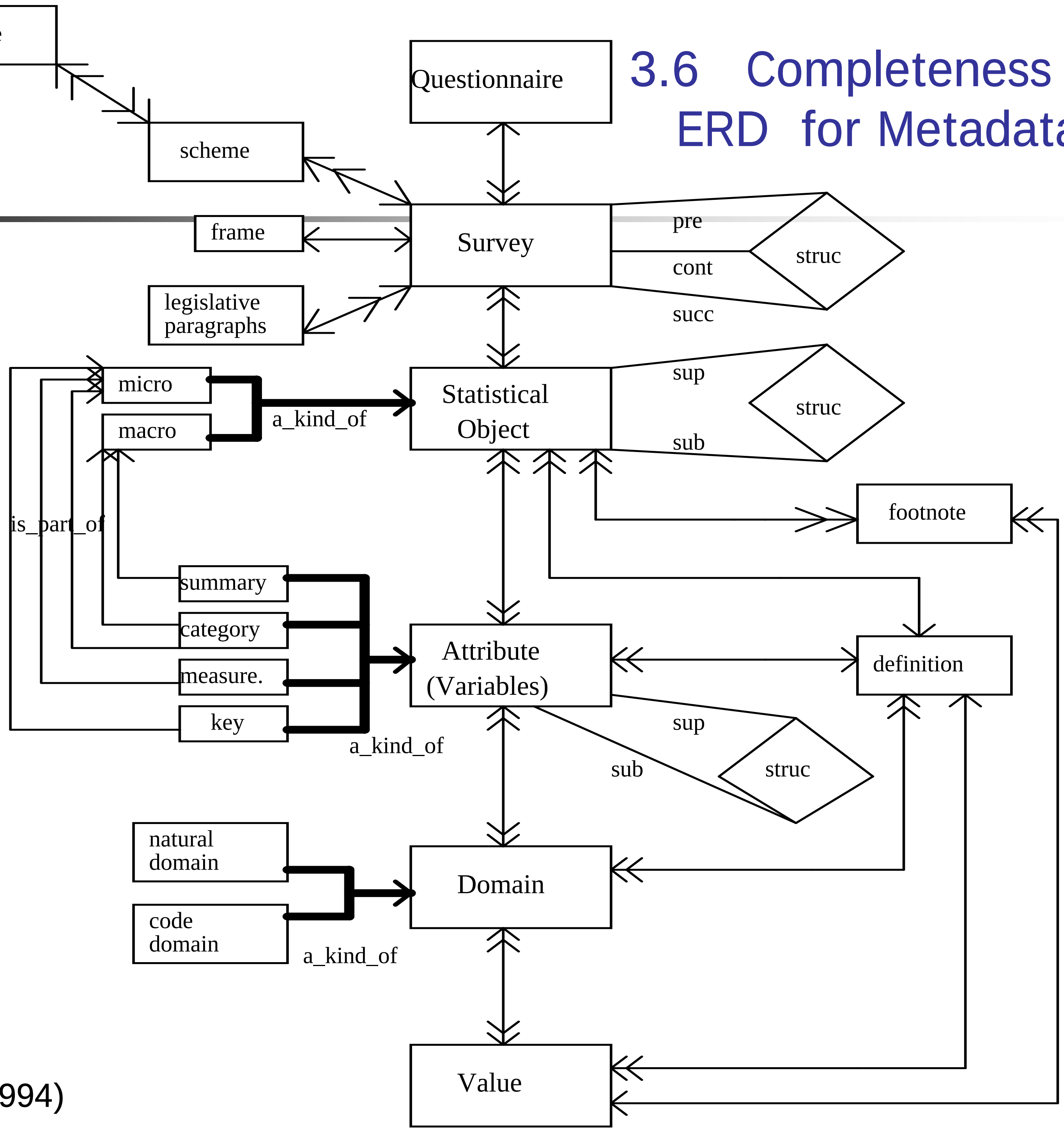
Person

<u>**Per#**</u>	**Firstname**	**Address**
1	Klaus	Seestr. 2
2	Hans	Garystr. 12
3	Otto	Heerstr. 10

Relation *Person* is not complete because attribute *FamilyName* is missing.

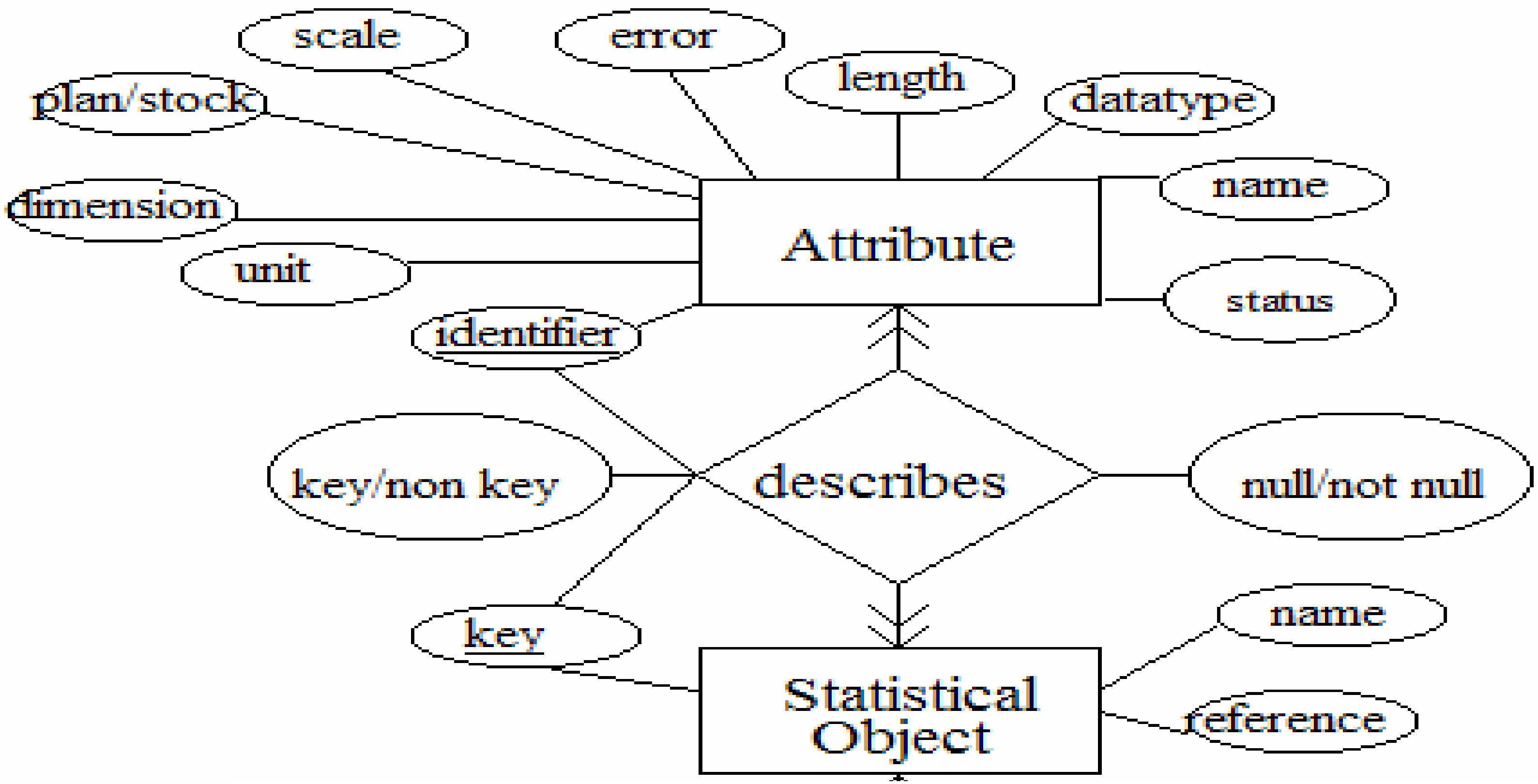
Quelle: C. Batini & M. Scannapieco: Data Quality – Concepts, Methodologies and Techniques, Springer, 2007

3.6 Completeness (cont.) ERD for Metadata



Source: Lenz(1994)

3.6 Completeness (cont.)
ERD for Metadata of entity „Attribute“



Sour

3.6 Completeness (cont.) Metadata

Metadata describe universes (populations), micro and macro data on the levels

- ***semantic,***
- ***structural,***
- ***statistical, and***
- ***physical***

in such a way that

- **the universe is well defined, and data can be reasonably**
- **inputted, stored, updated,**
- **transformed, grouped, summarized (aggregated),**
- **retrieved and disseminated**.

3.7 Pertinence („over modelling“)

- **Number of unnecessary objects included in the schema**

- **Ex:**

Person

Per#	**Firstname**	**FullName**	**Address**	**Hair Colour**
1	Klaus	Meier	Seestr. 2	brown
2	Hans	Müller	Garystr. 12	black
3	Otto	Schmidt	Heerstr.10	blond

„Over Modelling“:
„Hair colour“ is unnecessary for a citizen register
Note: Eye colour may be needed !

End of Schema Quality Dimensions

Good enough is not „good enough" !

4. Literatur

1. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methods and Techniques. Heidelberg: Springer Verlag (2006)
2. Dombrowski, Erik und Lechtenböger, Jens: Evaluation objektorientierter Ansätze zur Data-Warehouse-Modellierung, Datenbank-Spektrum 15/2005
3. Naumann, Felix: Datenqualität, Informatik-Spektrum_30_1_2007