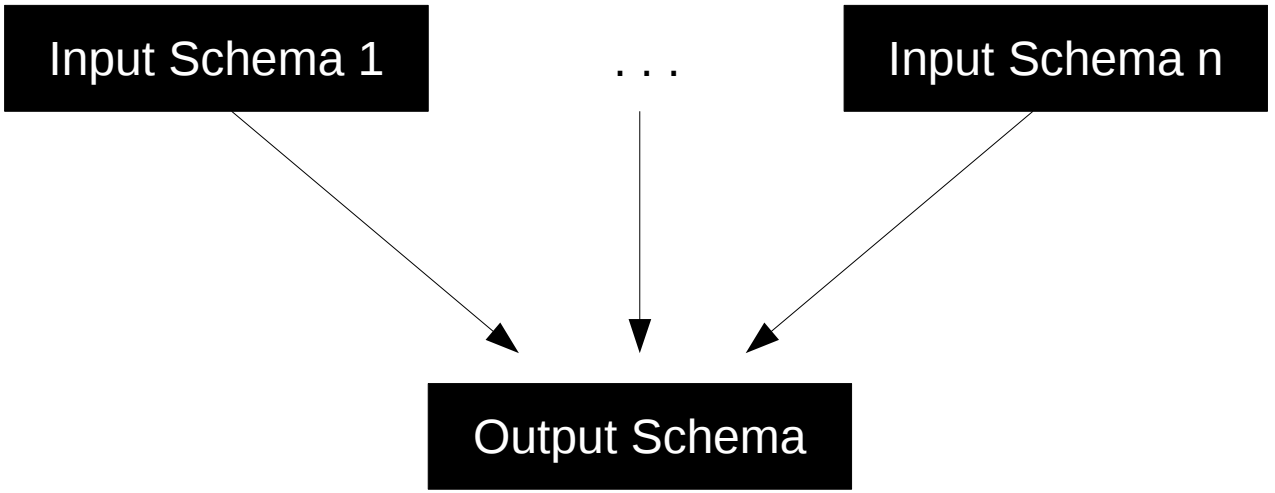
Logical Design
View Integration

Thomas Grafenauer
Stefan Petscharnig
Richard Taupe

What is schema integration?



The result cannot be proven!

The process is not formalized!

Goals of Schema Integration

- Merge heterogeneous schemas
- Result: integrated global schema
- Desirable properties:
  - Completeness
  - Minimality
  - Correctness
  - Comprehensibility

Why is it necessary?

- Huge designs done by more than one person
- Mergers & acquisitions
- New business areas
- Incremental development
- ...

Potential Problems

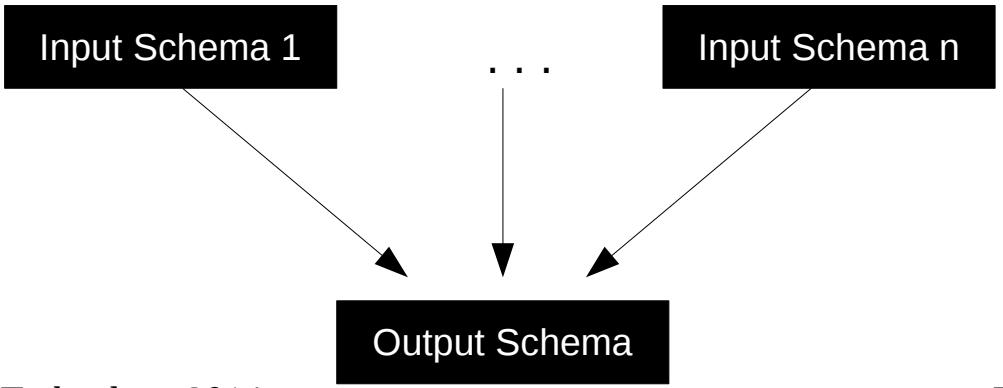
- Large schemas
- Unfamiliar schemas
- Heterogenous schemas
- Schemas in foreign languages
- Bad style (e. g. cryptic concept names)

Schema Integration Types

- One-step integration
- Incremental integration
- Mixed integration

One-step Integration

- Integrate all schemas in one step
- Also called „n-ary method“
- Advantage: No local decisions which might have to be discarded later
- Used if there are few / simple / small input schemas



Incremental Integration

- Integrate one at a time
- Also called „binary method“
- Advantages:
  - A single step is comparably easy (only two schemas are merged)
  - Schemas which are integrated earlier (e. g. because they are more important) have stronger effect on result
- Used if there are many input schemas

Mixed Integration

- First, integrate all schemas of same context
  - e. g. same department or product line
  - Use one-step or incremental method
- Then, integrate resulting intermediate schemas
  - Use one-step or incremental method

Integration Process

1) Conflict Analysis
2) Conflict Resolution
3) Schema Merging

Conflict Types 1/2

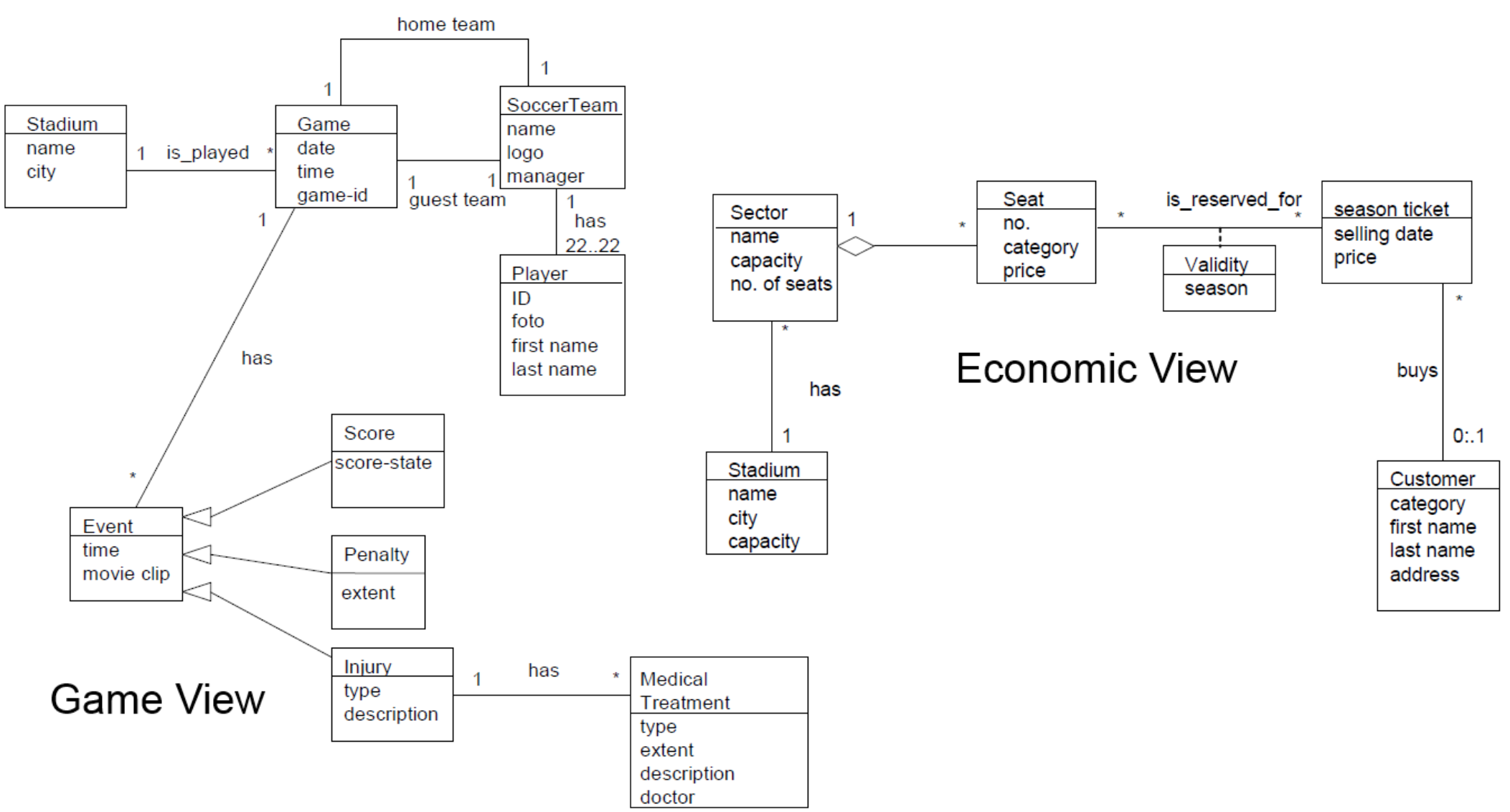
- Naming Conflicts
  - Homonyms: e. g. „menu“
    - Solution: rename in one schema
  - Synonyms: e. g. „buy“ / „purchase“, „film“ / „movie“
    - Solution: rename in one schema
  - Special cases: e. g. Same name but (partially) different neighbors and constraints
    - Maybe renaming or generalization works
    - Otherwise: Structural Conflict

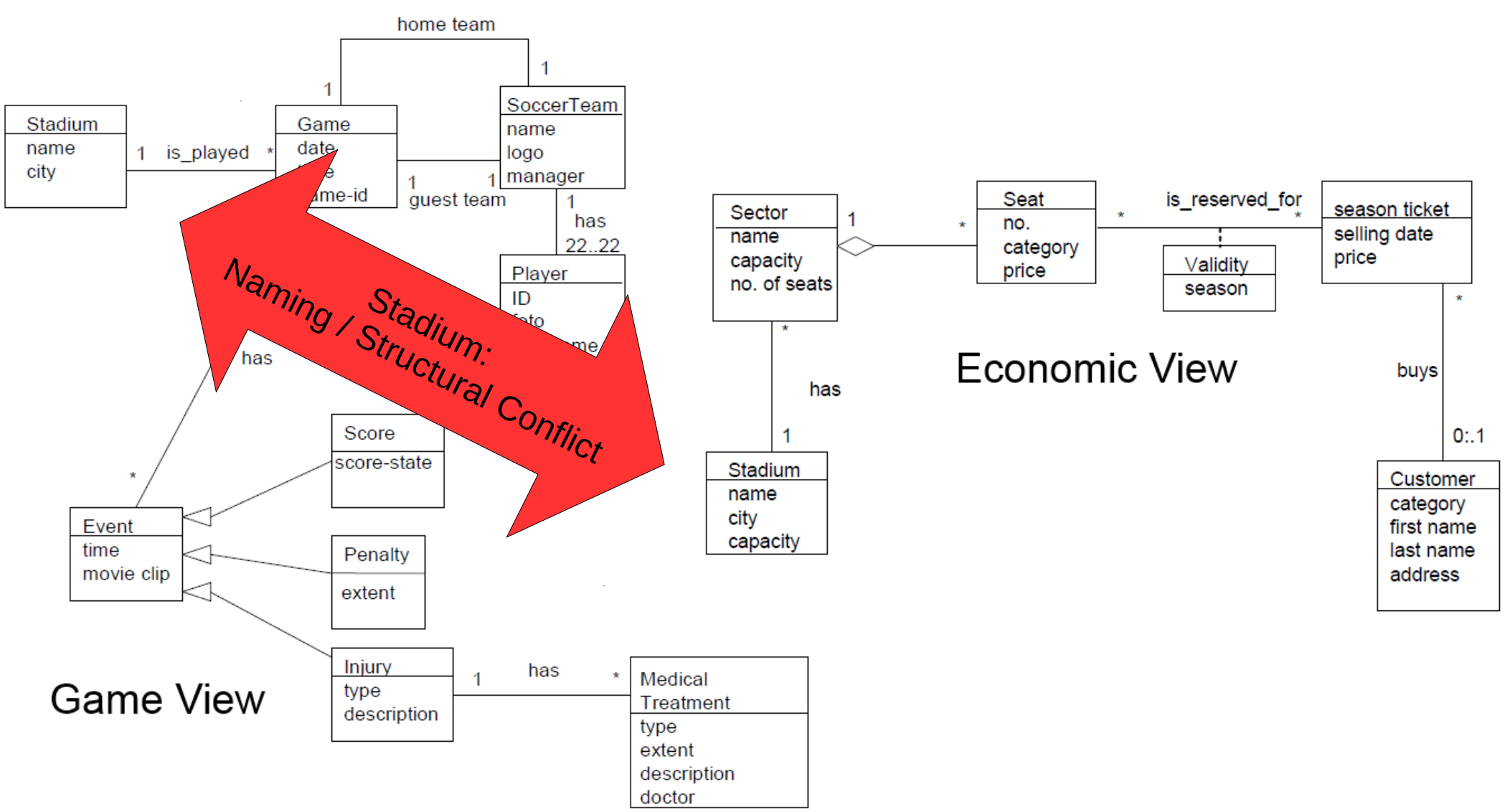
Conflict Types 2/2

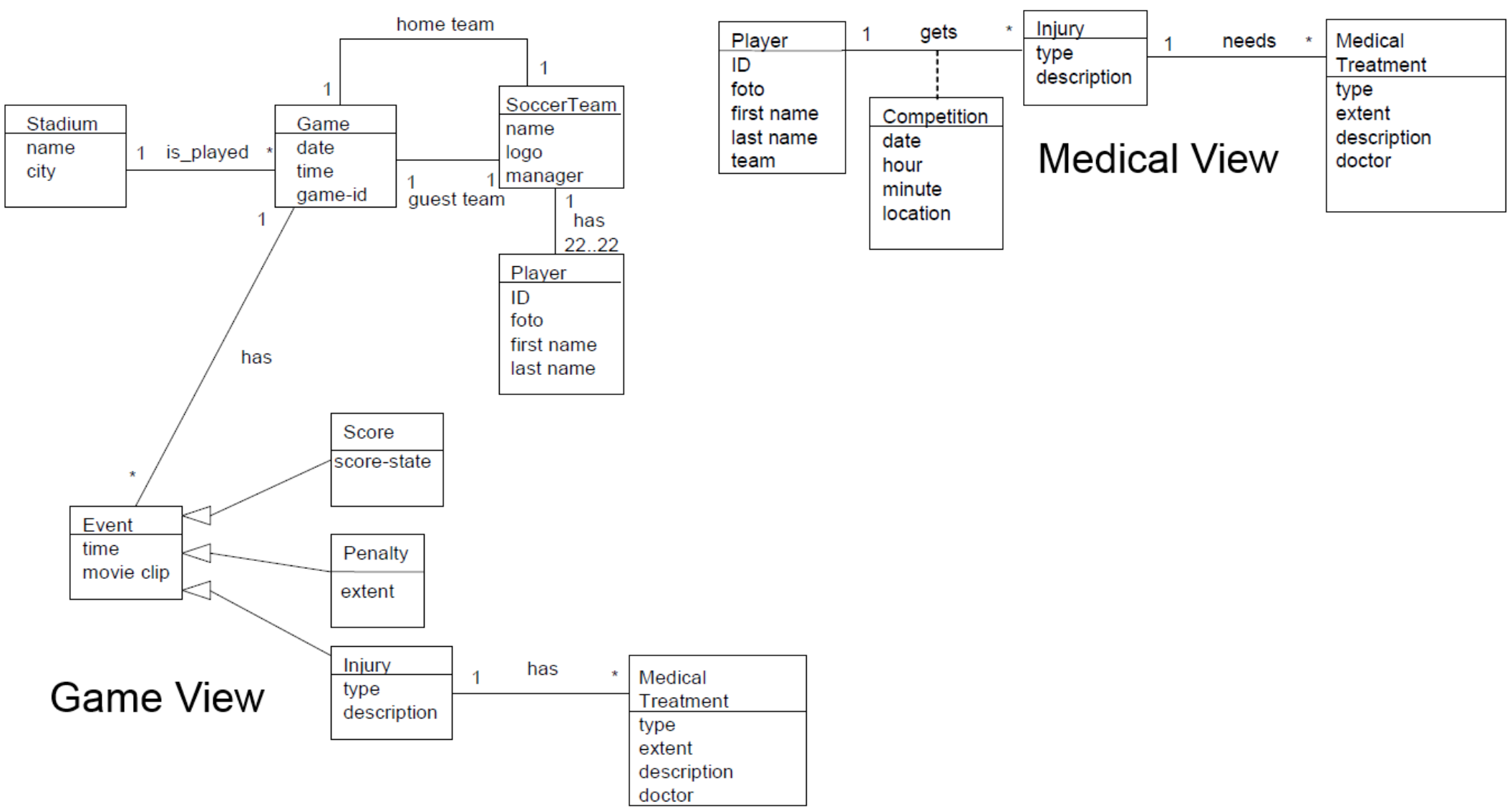
- Structural Conflicts
  - Identical concepts
    - Solution: do nothing
  - Compatible concepts
    - Example: Generalization vs. flattened generalization
    - Solution: adapt one input schema
  - Incompatible concepts
    - Examples: different multiplicities, types, …
    - Solution if design error: Adjust
    - Solution otherwise: Choose more general variant

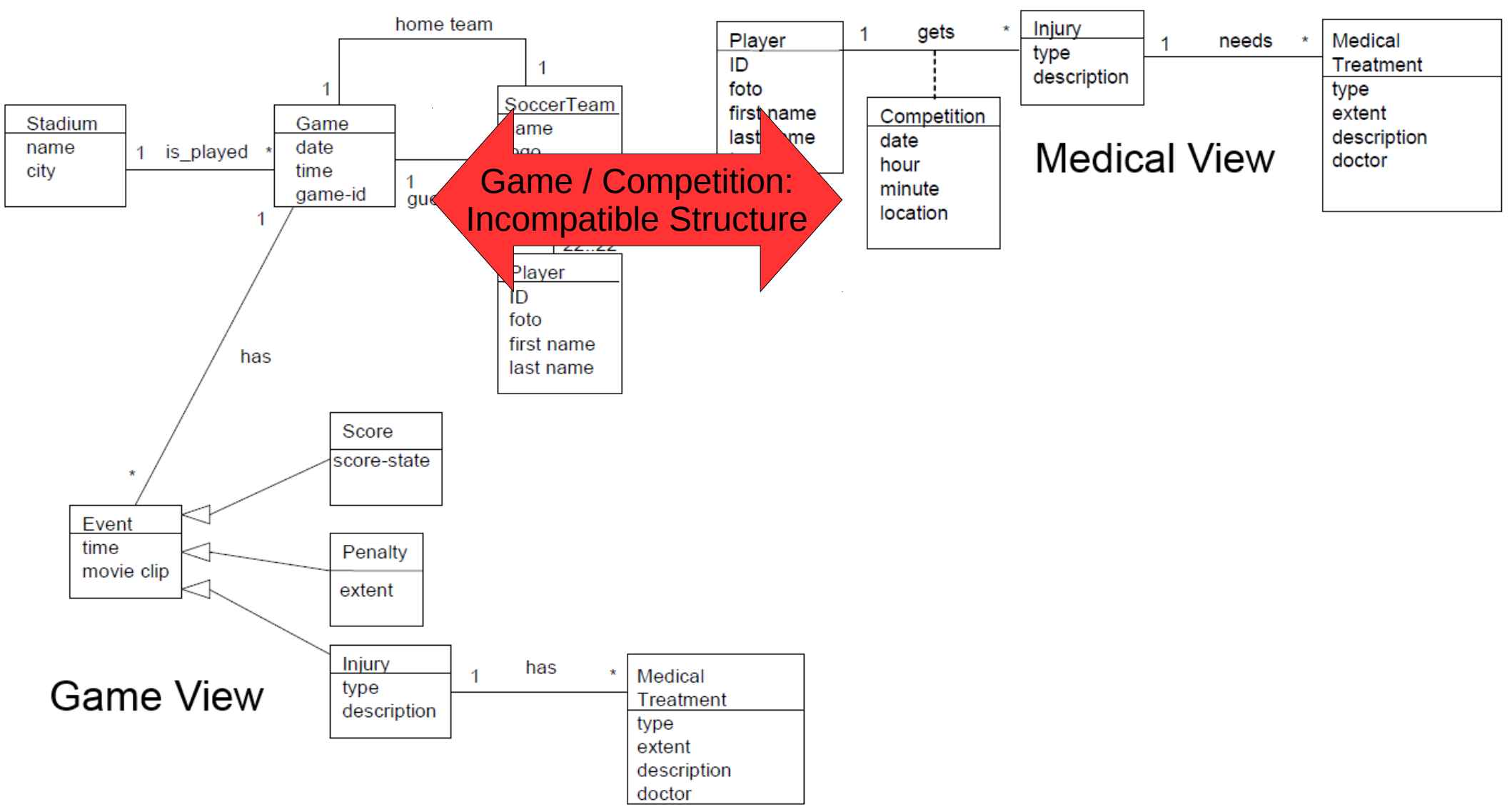
Exercise 1: View Integration

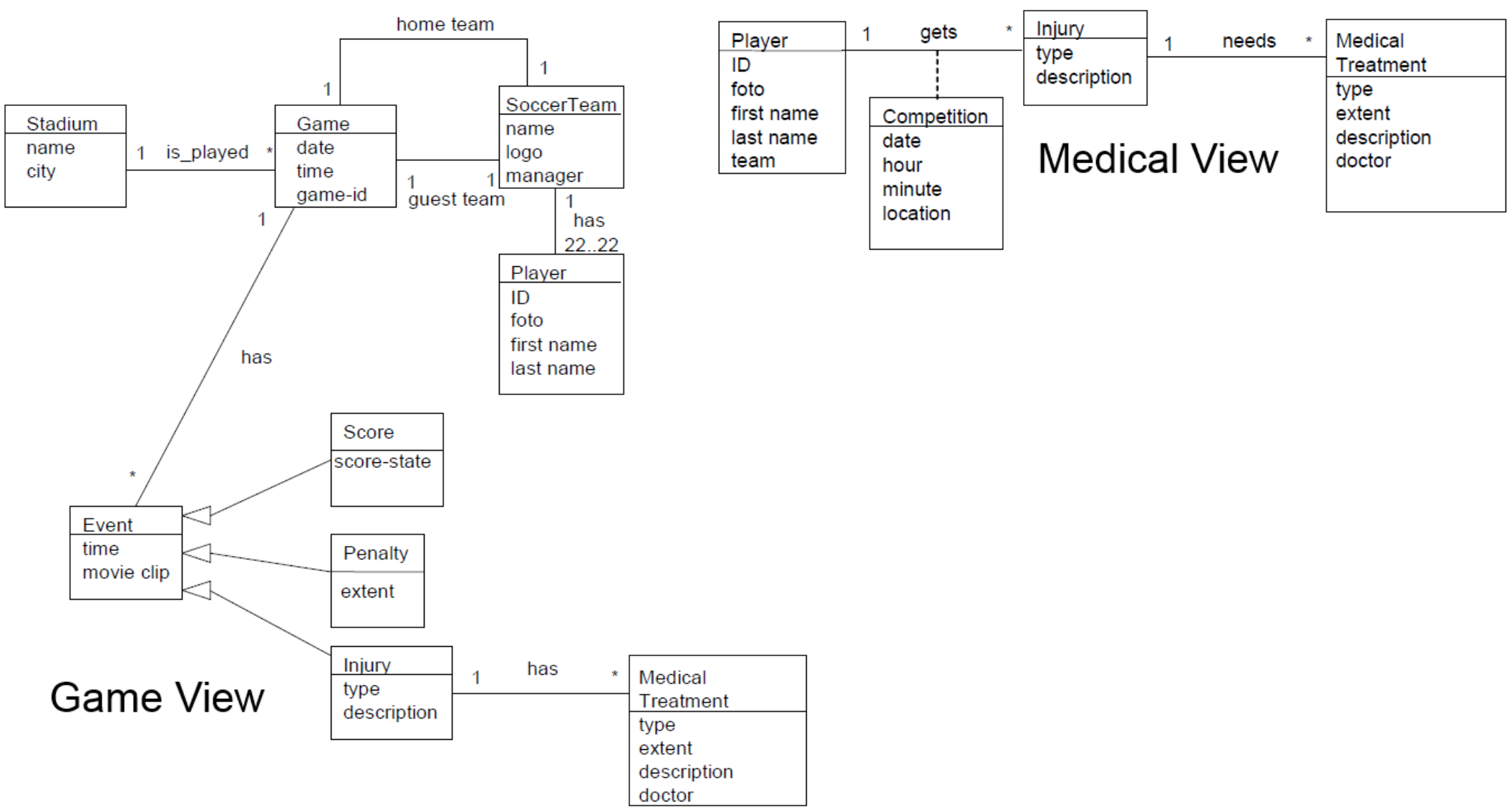
1) Identify conflicts
2) Propose scenarios and interschema properties
3) Integrate schemas

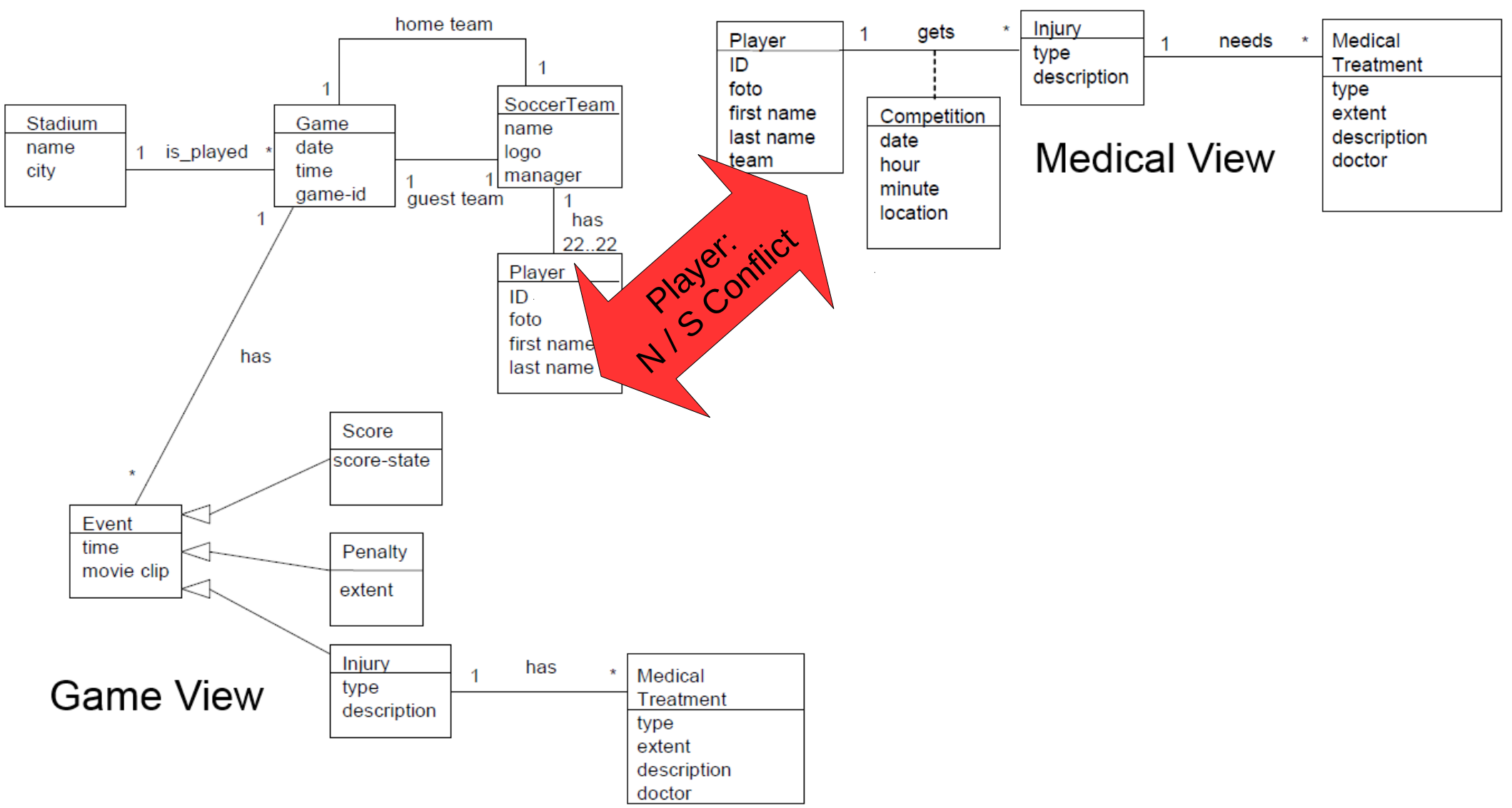

home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
1
1
guest team
1
SoccerTeam
name
logo
manager
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical Treatment
type
extent
description
doctor
Game View
Sector
name
capacity
no. of seats
1
*
Seat
no.
category
price
*
is_reserved_for
*
Validity
season
season ticket
selling date
price
*
buys
0:.1
Customer
category
first name
last name
address
*
has
1
Stadium
name
city
capacity
Economic View

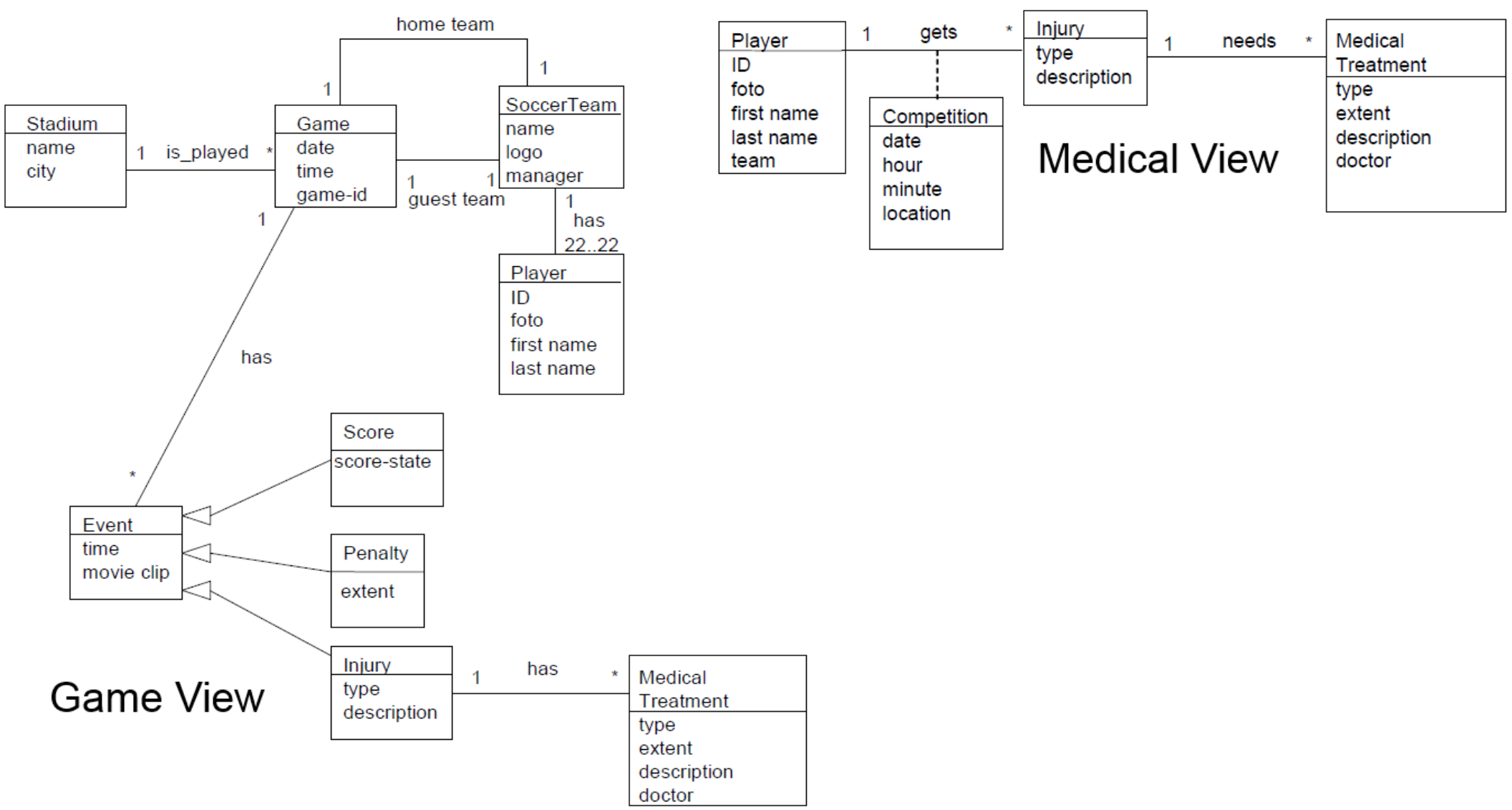

home team
1
1
Stadium
name
city
1
is_played
*
Game
date
me-id
1
guest team
1
SoccerTeam
name
logo
manager
1
has
22..22
Player
ID
me
has
*
Stadium:
Naming / Structural Conflict
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View
Sector
name
capacity
no. of seats
1
*
Seat
no.
category
price
*
is_reserved_for
*
Validity
season
season ticket
selling date
price
*
has
1
Stadium
name
city
capacity
buys
0:.1
Customer
category
first name
last name
address
Economic View

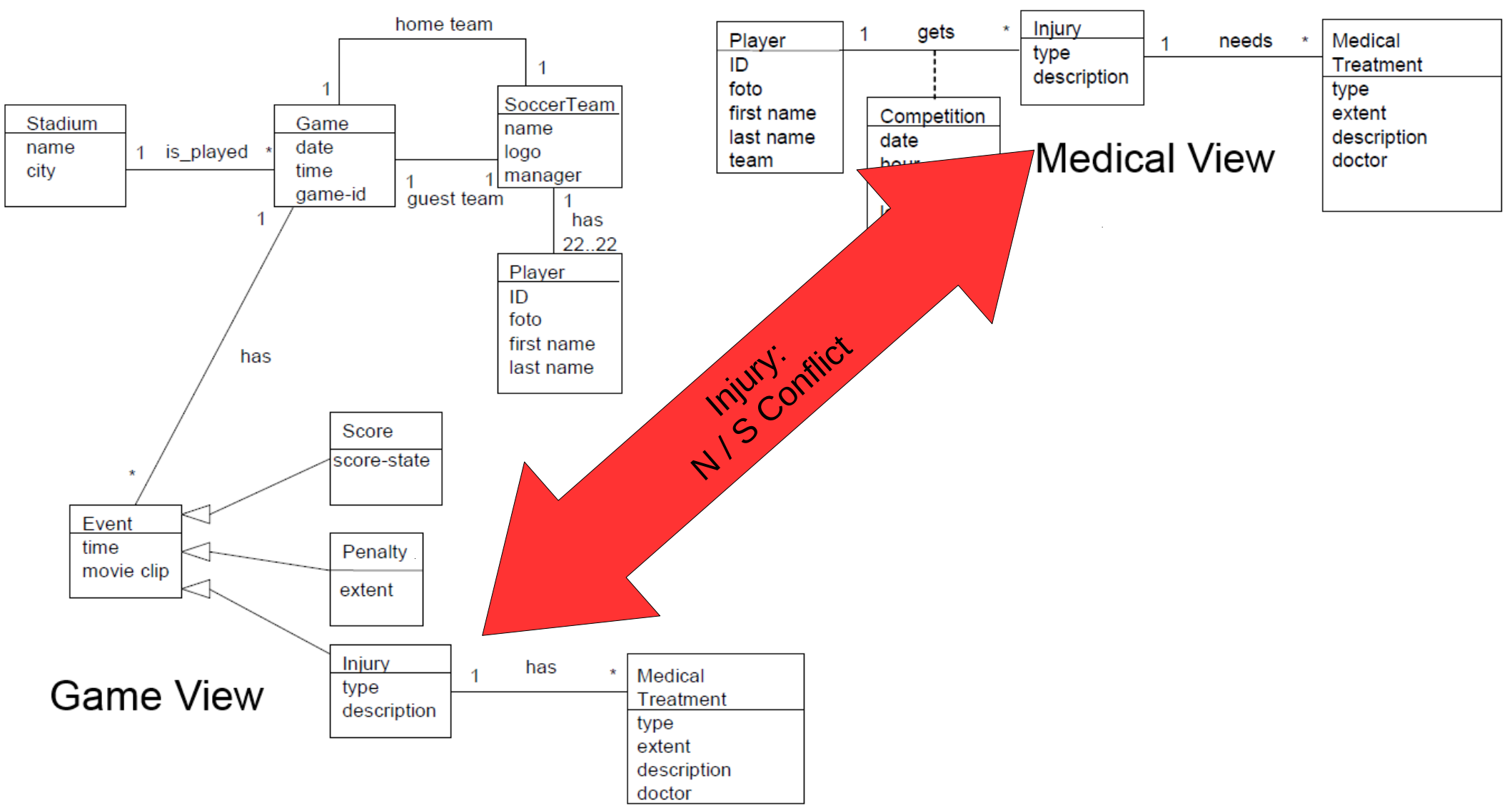

home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
1
guest team
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical
Treatment
type
extent
description
doctor
Medical View

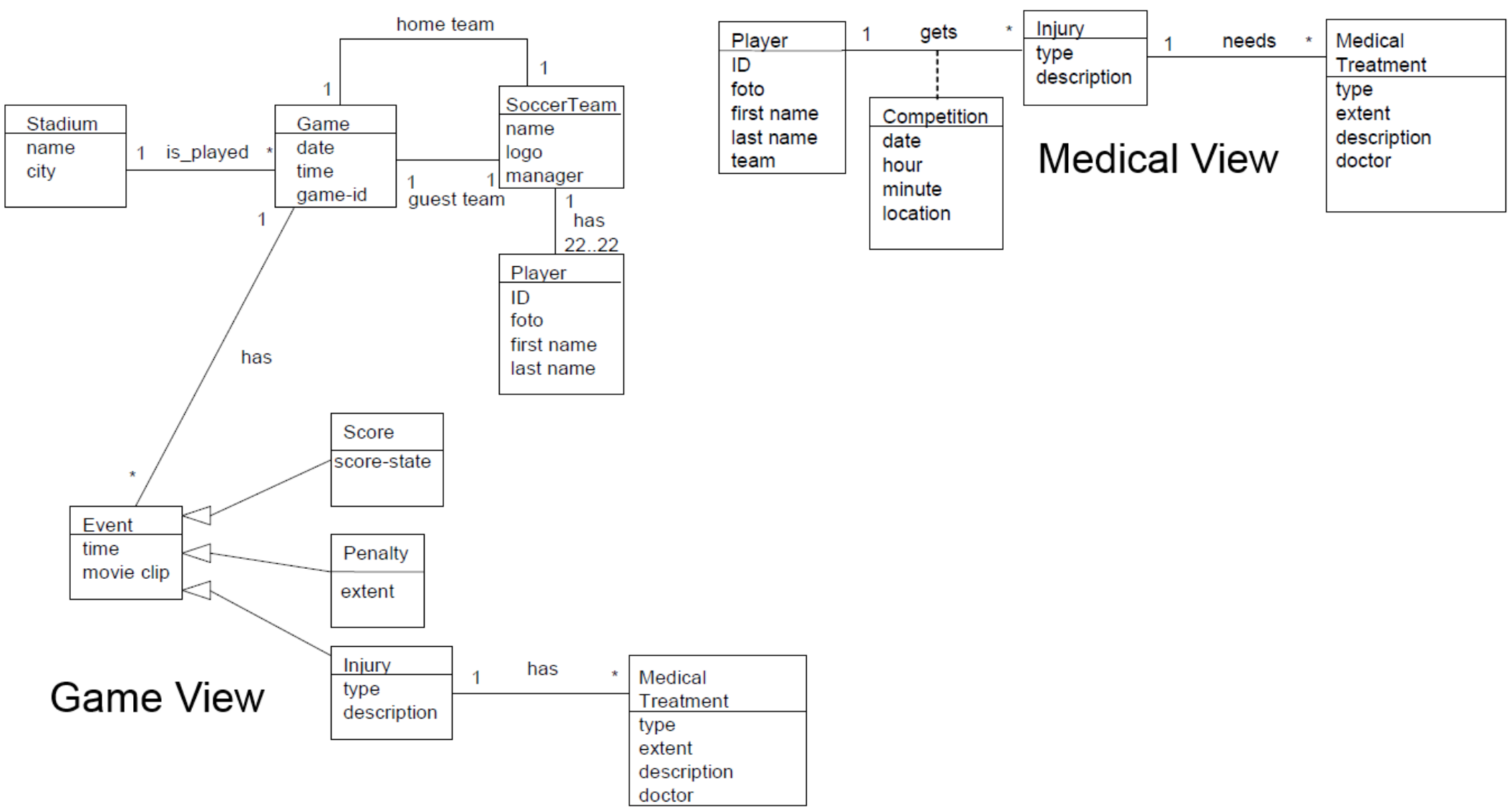

home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
1
gue
SoccerTeam
ame
Player
ID
foto
first name
last name
Game / Competition:
Incompatible Structure
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical
Treatment
type
extent
description
doctor
Medical View
1
has
*
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View

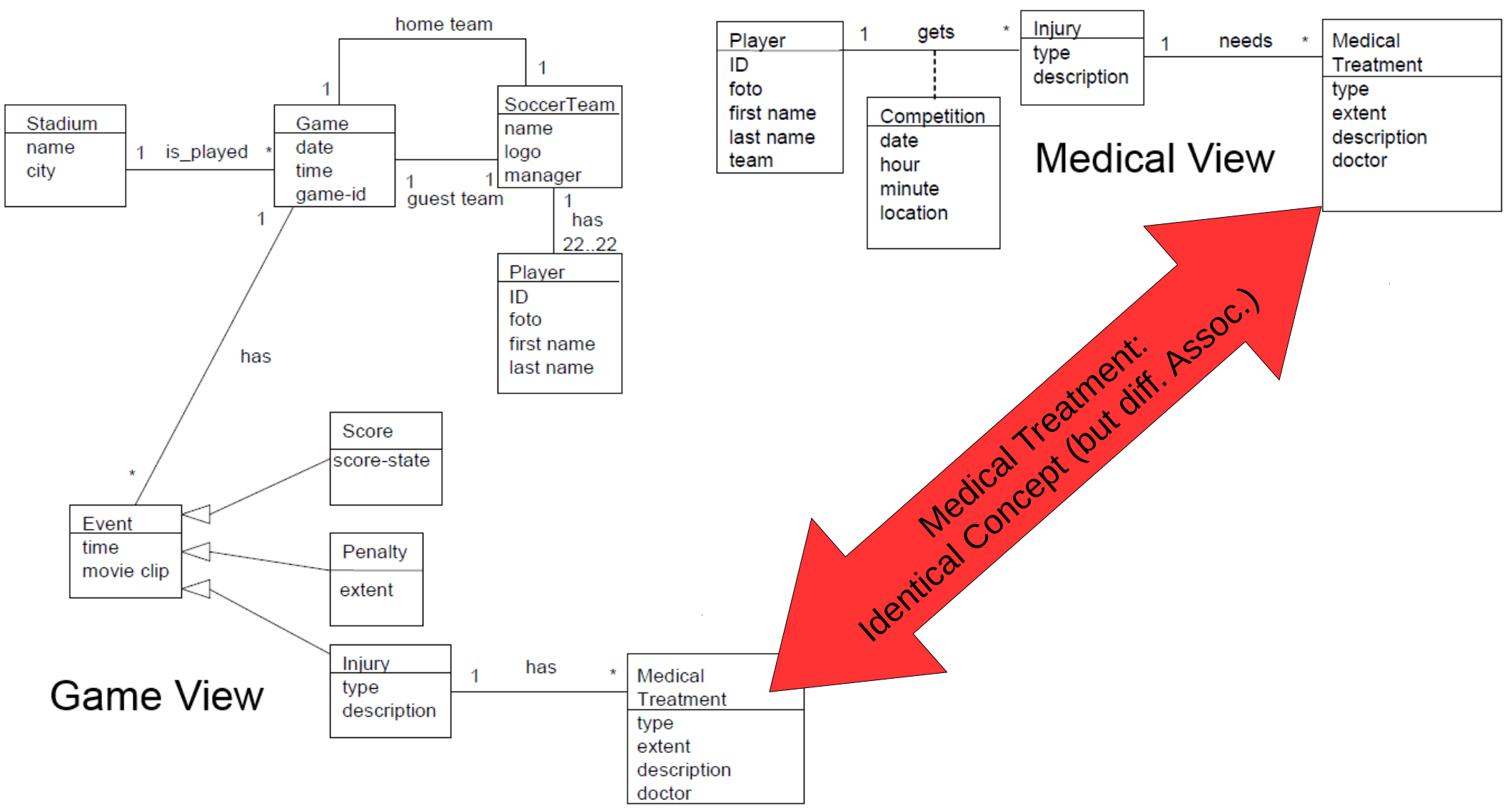

home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
1
guest team
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical
Treatment
type
extent
description
doctor
Medical View


home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
1
guest team
1
has
22..22
1
has
*
Player
ID
foto
first name
last name
Player:
N / S Conflict
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical
Treatment
type
extent
description
doctor
Medical View
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View


home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
guest team
1
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical Treatment
type
extent
description
doctor
Game View
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical Treatment
type
extent
description
doctor
Medical View


home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
1
guest team
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical Treatment
type
extent
description
doctor
Game View
Player
ID
foto
first name
last name
1
gets
*
Competition
date
Injury
type
description
1
needs
*
Medical Treatment
type
extent
description
doctor
Medical View
Injury:
N / S Conflict

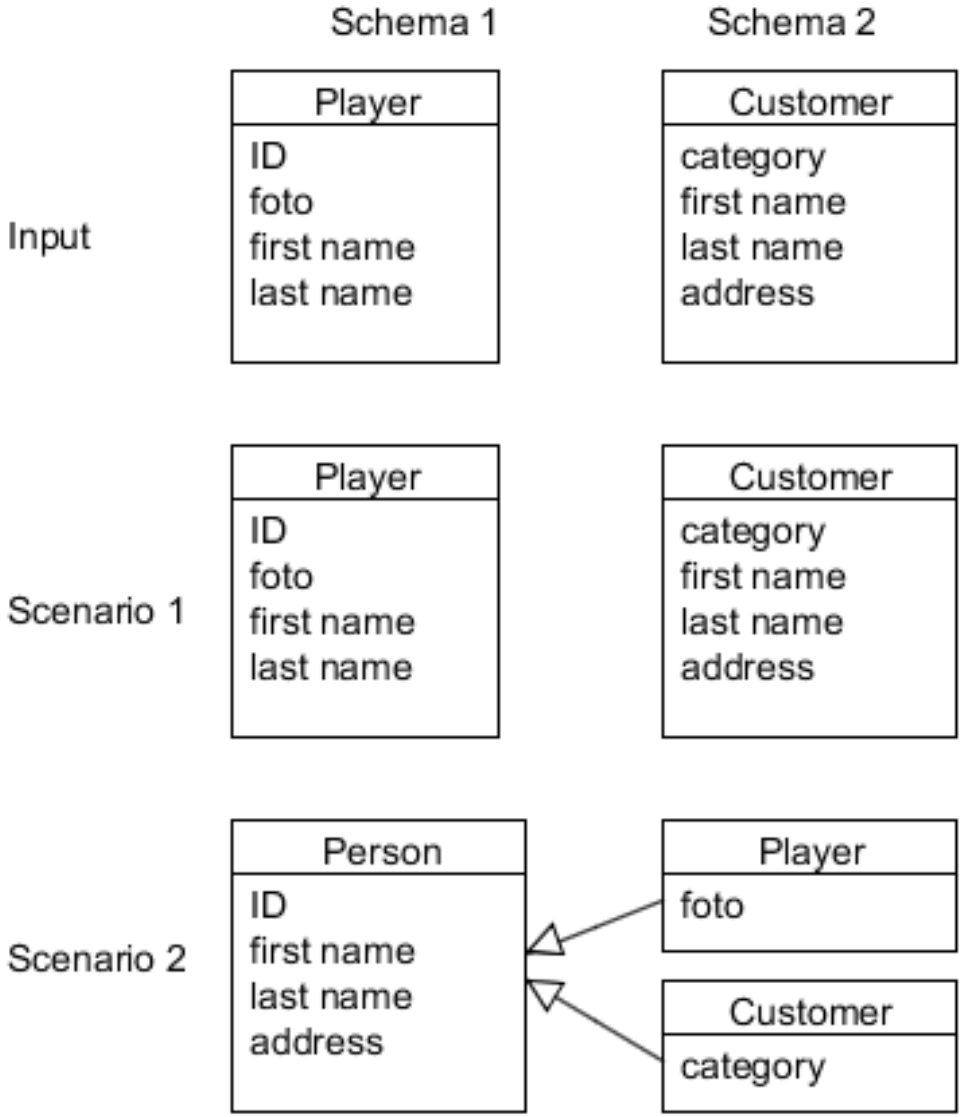

home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
1
guest team
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical
Treatment
type
extent
description
doctor
Medical View


home team
1
1
Stadium
name
city
1
is_played
*
Game
date
time
game-id
SoccerTeam
name
logo
manager
1
1
guest team
1
1
has
22..22
Player
ID
foto
first name
last name
1
has
*
Score
score-state
Event
time
movie clip
Penalty
extent
Injury
type
description
1
has
*
Medical
Treatment
type
extent
description
doctor
Game View
Player
ID
foto
first name
last name
1
gets
*
Competition
date
hour
minute
location
Injury
type
description
1
needs
*
Medical
Treatment
type
extent
description
doctor
Medical View
Medical Treatment:
Identical Concept (but diff. Assoc.)

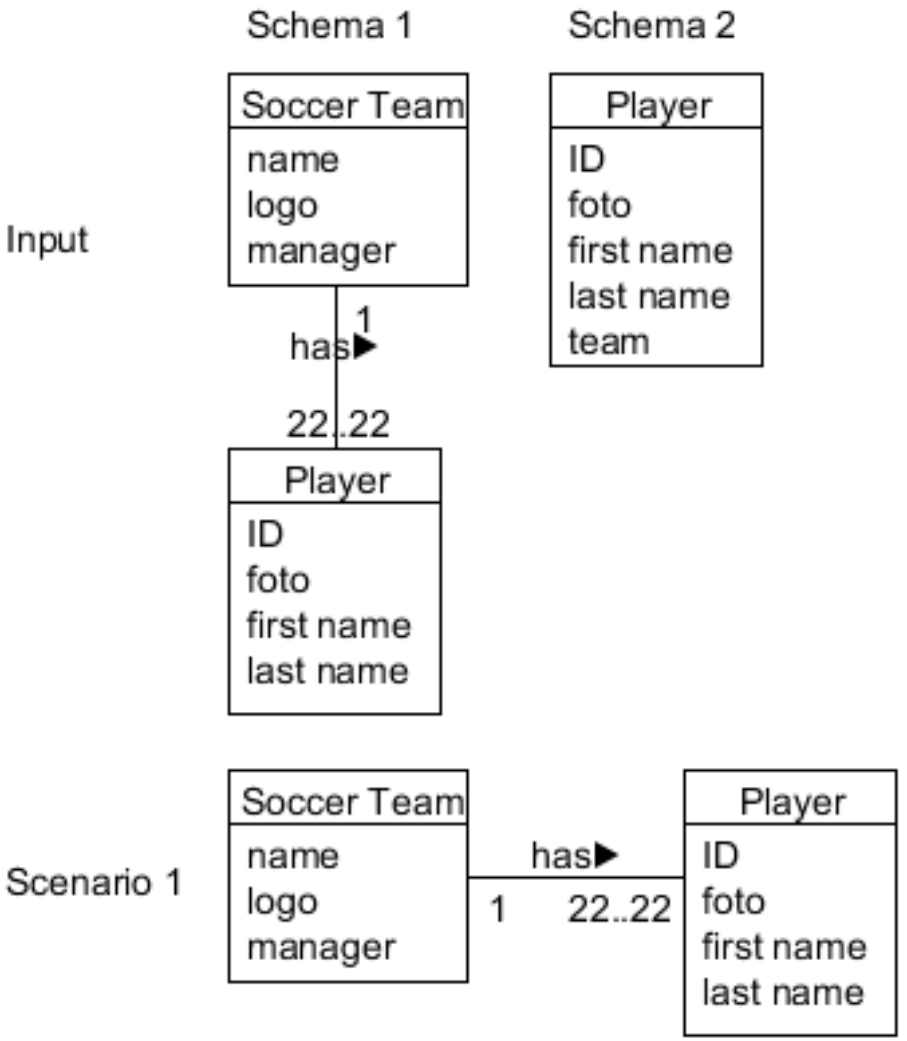
Exercise 1: View Integration

1) ~~Identify conflicts~~
2) Propose scenarios and interschema properties
3) Integrate schemas

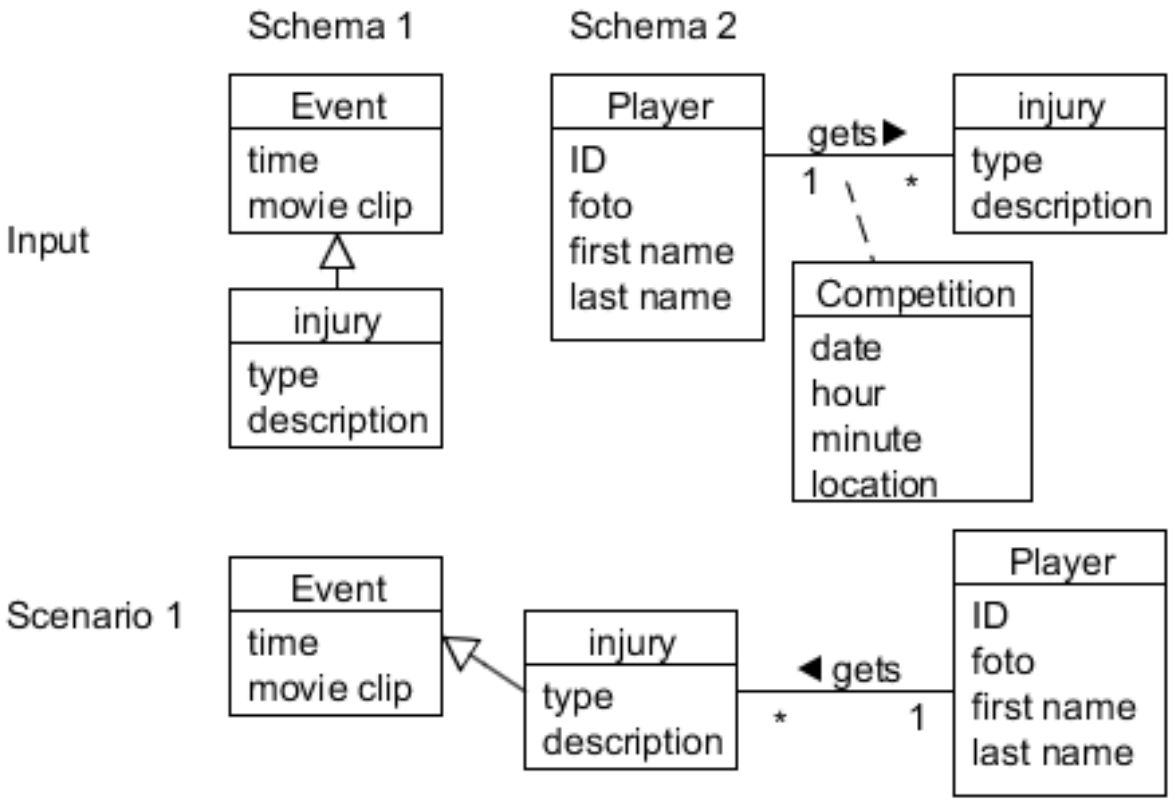
Interschema Properties & Scenarios



Interschema Properties & Scenarios



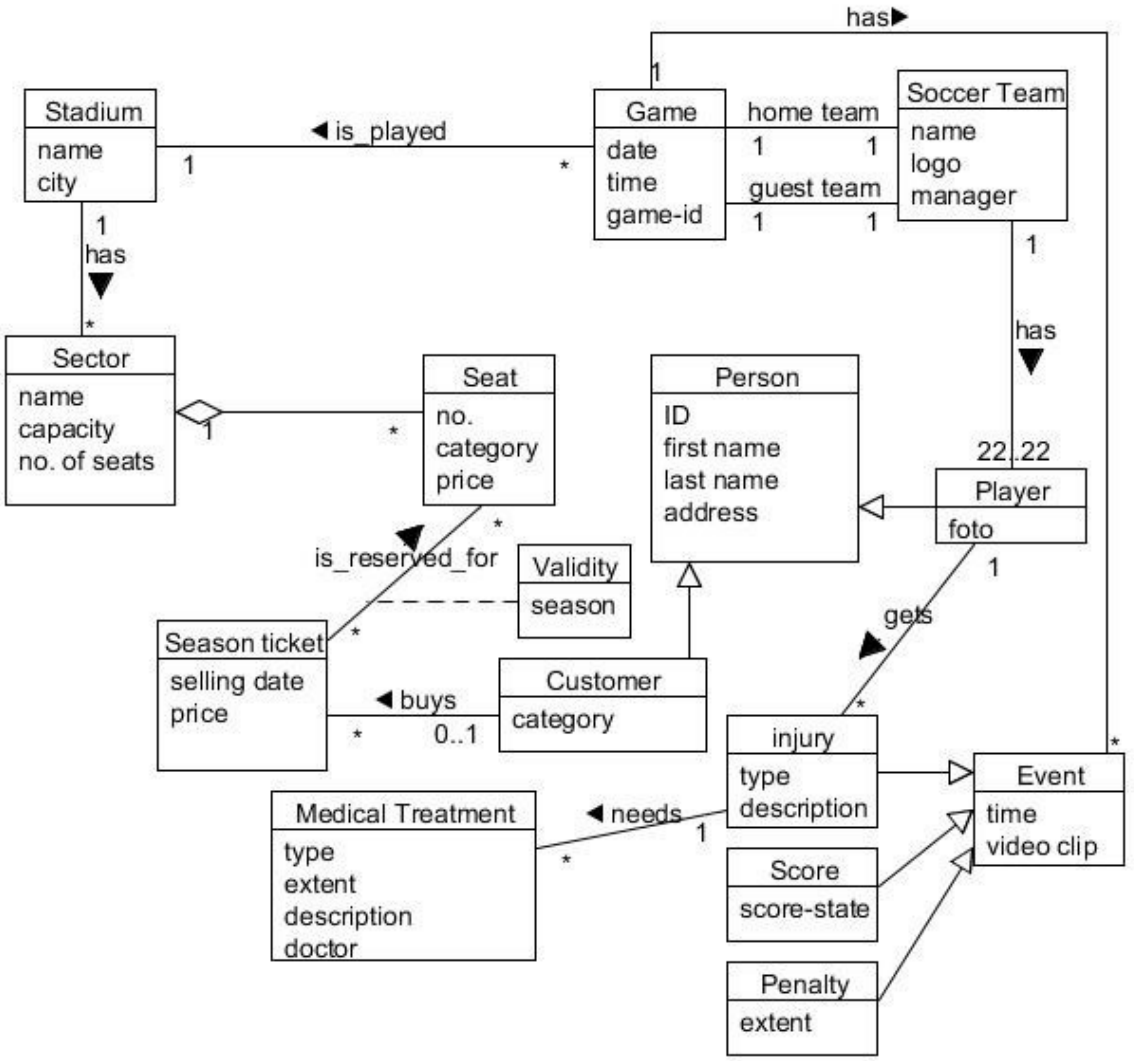
Interschema Properties & Scenarios



Exercise 1: View Integration

1) ~~Identify conflicts~~
2) ~~Propose scenarios and interschema properties~~
3) Integrate schemas

Final Integrated Schema



Exercise 1: View Integration

1) ~~Identify conflicts~~
2) ~~Propose scenarios and interschema properties~~
3) ~~Integrate schemas~~

Data-Volume-Quantities

- To gain knowledge about boundaries
  - min entries for a table
  - avg entries for a table
  - max entries for a table
  - avg.growth for a table
- Valuable for performance calculations
  - How many reads/writes occur in average?

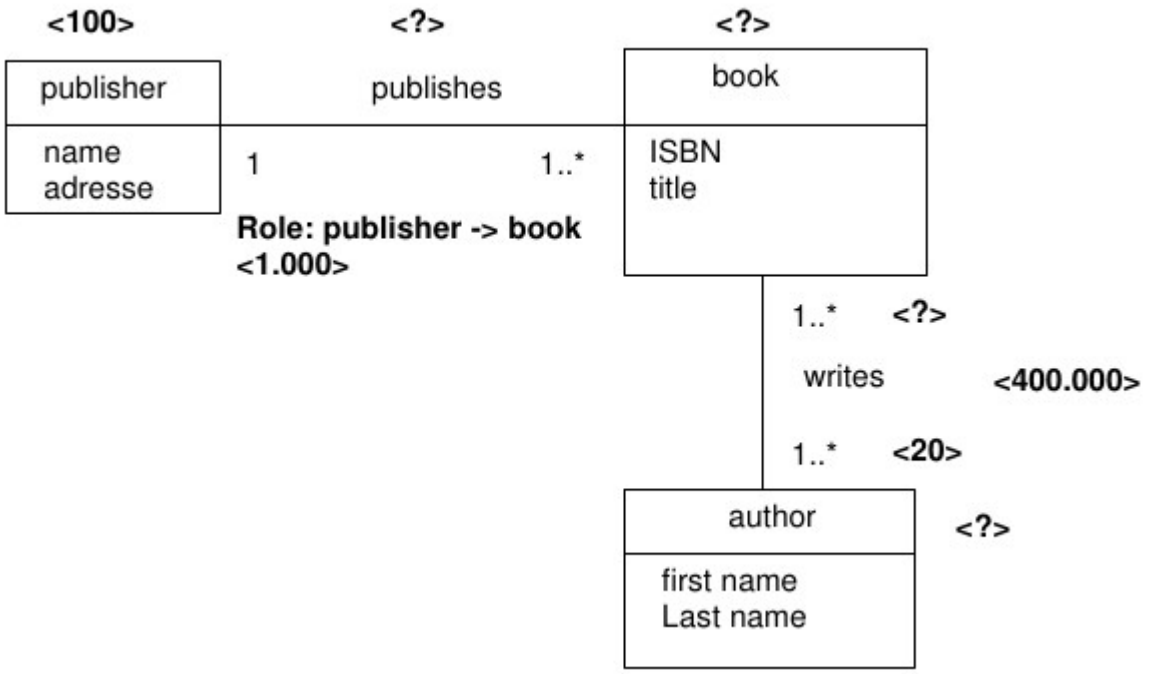
Data-Volume-Information

- Data-Volume-Information:
  - N(C): avg. number of instances per class
  - N(A): avg. number of instances per association
  - N(C,A): avg. cardinality of roles
- Attribute cardinality
  - Cardinality of attributes domain
  - e.g. one person can have multiple telephone-numbers

Determination of Quantities

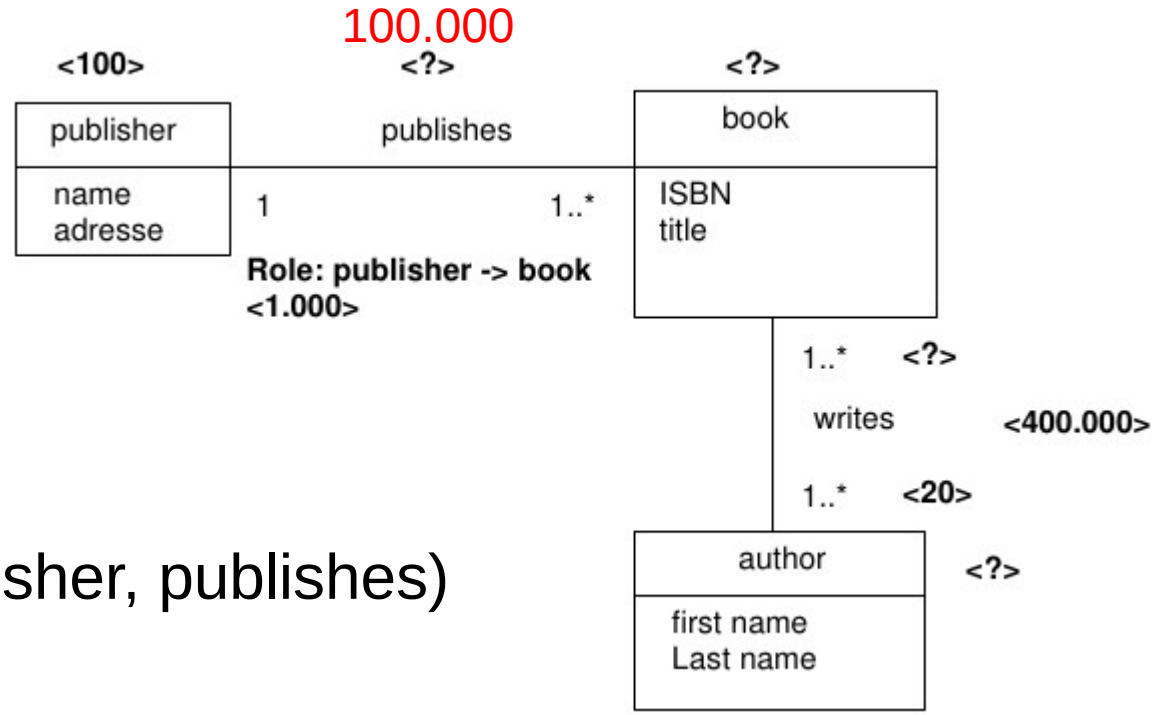
- Classes: C1, C2
- Association A
- Roles: (C1,A) and (C2,A)
- N(C1) x N(C1,A) = N(A) = N(C2) x N(C2,A)

Example


<100>
<?>
<?>
publisher
publishes
book
name
adresse
1
1..*
ISBN
title
Role: publisher -> book
<1.000>
1..*
<?>
writes
<400.000>
1..*
<20>
author
<?>
first name
Last name

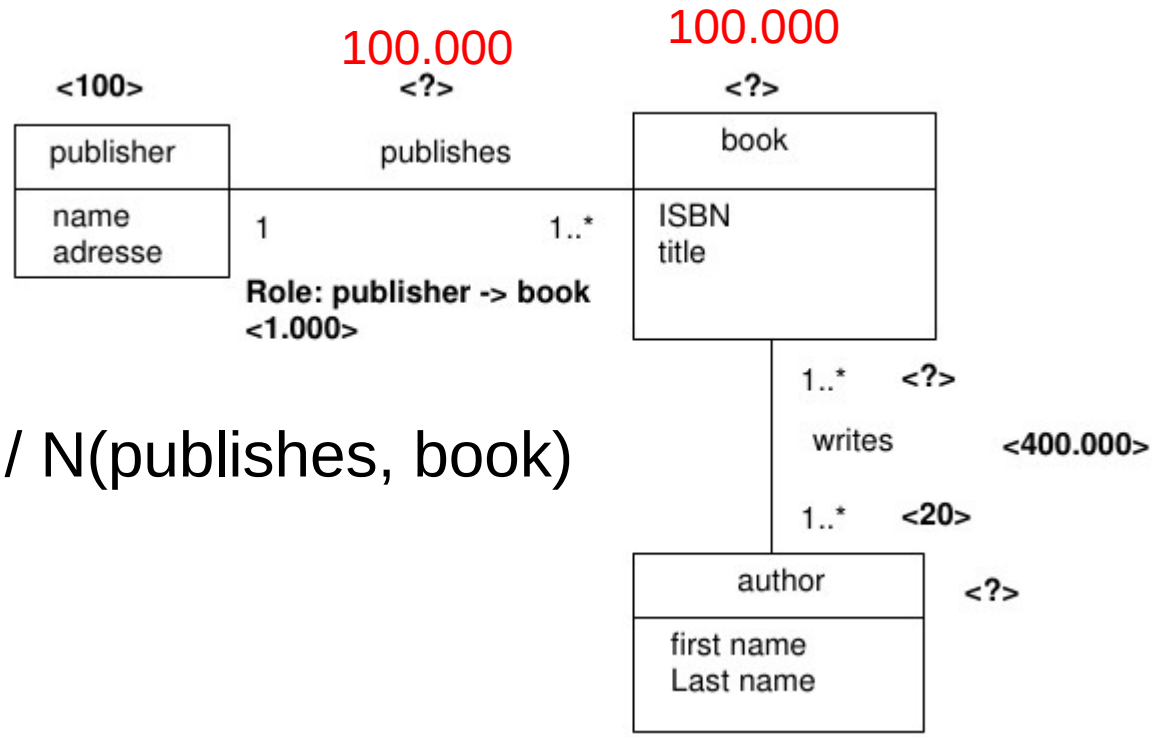
Example

- 1 publisher publishes 1000 books
- There are 100 publisher
- Means:
  N(publishes)
  = N(publisher) x N(publisher, publishes)
  = 100 x 1.000
  = 100.000



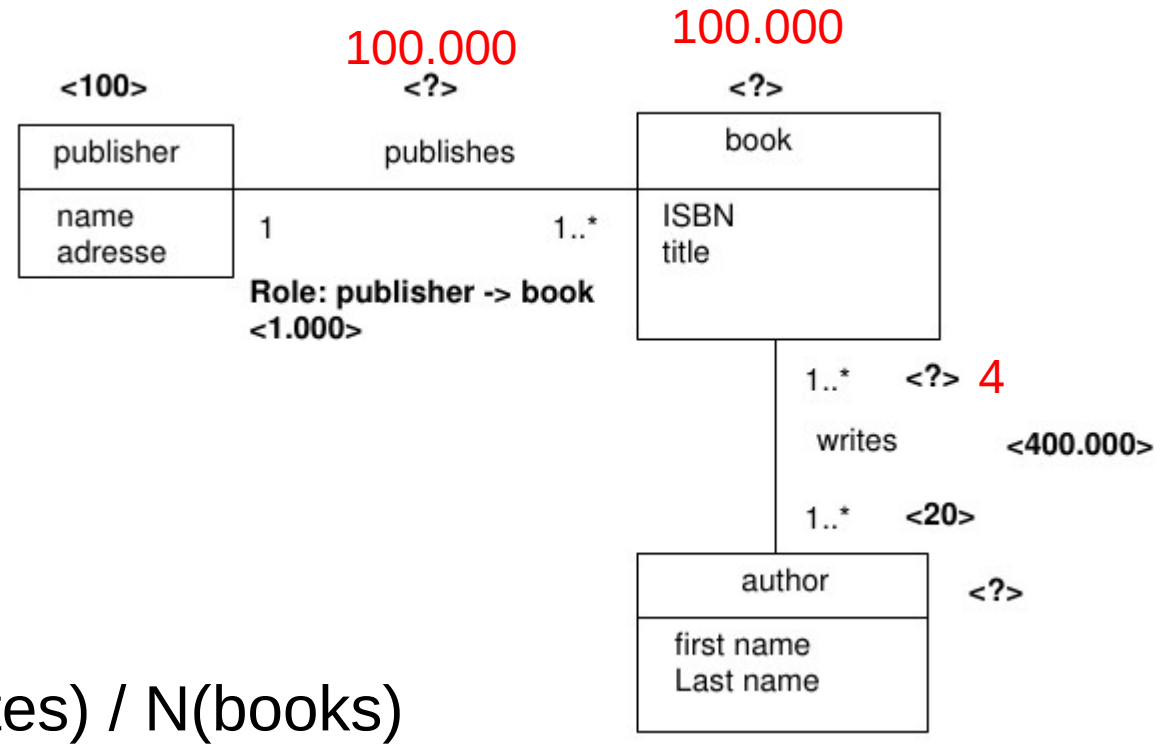
Example

- 1 book can be published by only 1 publisher
- N(book) = N(publishes) / N(publishes, book)
  = 100.000 / 1
  = 100.000



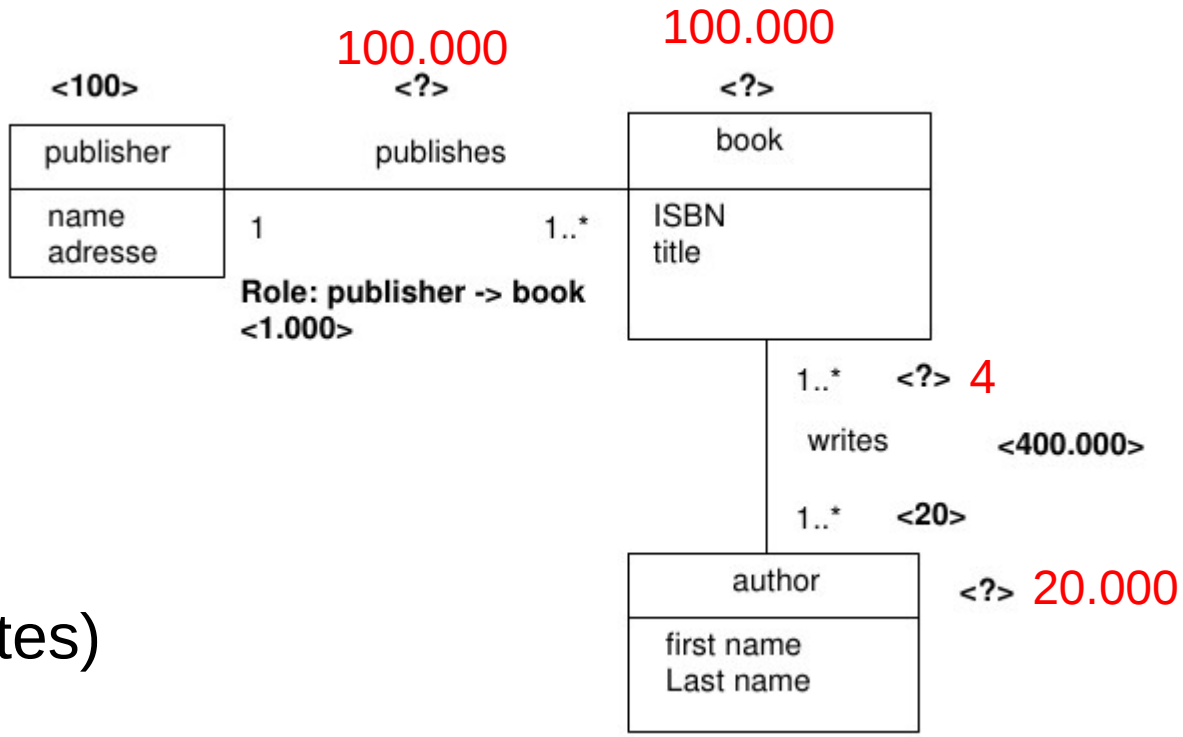
Example

- 100.000 books are in 400.000 writes-relations
- So 1 book is in 4 write-relations on average
- N(book, writes) = N(writes) / N(books) = 400.000 / 100.000 = 4



Example

- 1 authors is
  in 20 writes-relations
  on average
- How many authors?
- N(authors) =
  N(writes) / N(author, writes)
  = 400.000 / 20 = 20.000



Navigation Path

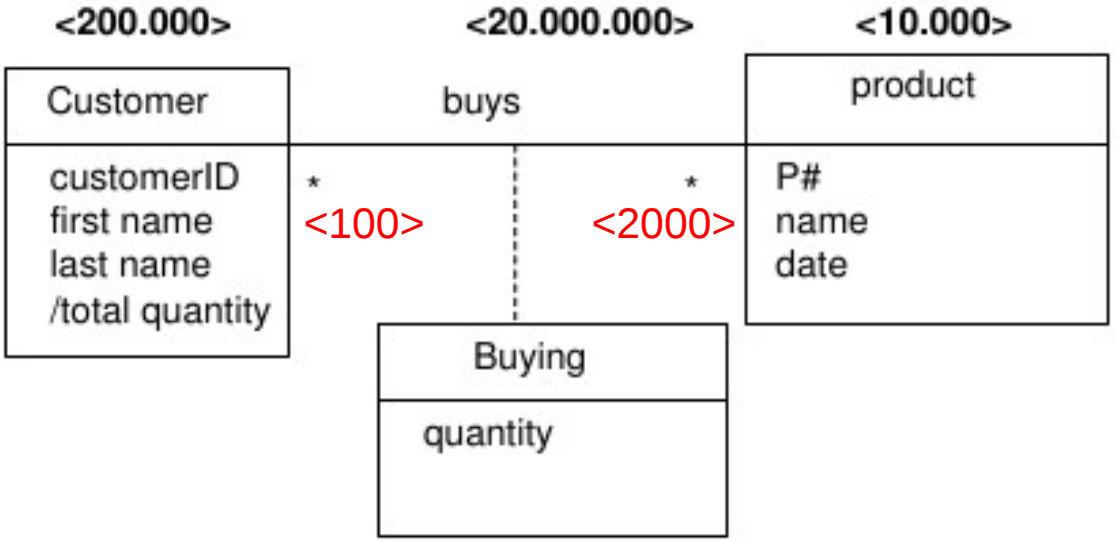
- Path through database for a query
- Amount of joins within a path
- Necessary for calculating cost of a query

Redundancy in Databases

- Need more memory-space
- Poor write-performance
- Can lead to inconsistency → anomalies
- Hence should be avoided → normalization
- Why let redundancy occur?
  - Avoidance is more expensive than problems
    e.g. surnames (Mrs. Jackson and Mr. Jackson)
  - Better read-performance because of less joins

Redundancy Example

- Average quantities
- A month = 20 working-days
- Weights:
  - read = 1
  - write = 2
  - update = 3
- Introduce attribute “total quantity” ?



Redundancy Example

- Transaction T1: Insert new customer → 50 times per month

	With „total quantity“	Without „total quantity“
sum	**0** access-costs	**0** access-costs

Redundancy Example

- Transaction T1: Insert new customer → 50 times per month

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
sum	**100** access-costs	**100** access-costs

Redundancy Example

- Transaction T2: Insert new product → 5 times per month

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
sum	**100** access-costs	**100** access-costs

Redundancy Example

- Transaction T2: Insert new product → 5 times per month

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
T2	5 write-accesses on product = 5*2 = **10**	5 write-accesses on product = 5*2 = **10**
sum	**110** access-costs	**110** access-costs

Redundancy Example

- Transaction T3: Insert in “Buying → 50 times per day

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
T2	5 write-accesses on product = 5*2 = **10**	5 write-accesses on product = 5*2 = **10**
sum	**110** access-costs	**110** access-costs

Redundancy Example

- Transaction T3: Insert in “Buying → 50 times per day

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
T2	5 write-accesses on product = 5*2 = **10**	5 write-accesses on product = 5*2 = **10**
T3	50 write-accesses/day = 20*50/month = 1000 * 2 = **2000** on buying +1000 update-accesses on customer = 1000*3 = **3000**	50 write-accesses/day = 20*50/month = 1000 * 2 = **2000** on buying
sum	**5.110** access-costs	**2.110** access-costs

Redundancy Example

- Transaction T4: Listing total quantity → 2 times per month

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
T2	5 write-accesses on product = 5*2 = **10**	5 write-accesses on product = 5*2 = **10**
T3	50 write-accesses/day = 20*50/month = 1000 * 2 = **2000** on buying +1000 update-accesses on customer = 1000*3 = **3000**	50 write-accesses/day = 20*50/month = 1000 * 2 = **2000** on buying
sum	**5.110** access-costs	**2.110** access-costs

Redundancy Example

- Transaction T4: Listing total quantity → 2 times per month

	With „total quantity“	Without „total quantity“
T1	50 write-accesses on customer = 50*2 = **100**	50 write-accesses on customer = 50*2 = **100**
T2	5 write-accesses on product = 5*2 = **10**	5 write-accesses on product = 5*2 = **10**
T3	50 write-accesses/day = 20*50/month = 1000 * 2 = **2000** on buying +1000 update-accesses on customer = 1000*3 = **3000**	50 write-accesses/day = 20*50/month = 1000 * 2 = **2000** on buying
T4	200.000 read-accesses on customer = 1*200.000 2-times per month = **400.000**	20.000.000 read-accesses on buying = 1*20.000.000 2-times per month = **40.000.000**
sum	**405.110** access-costs	**40.002.110** access-costs

Redundancy Example

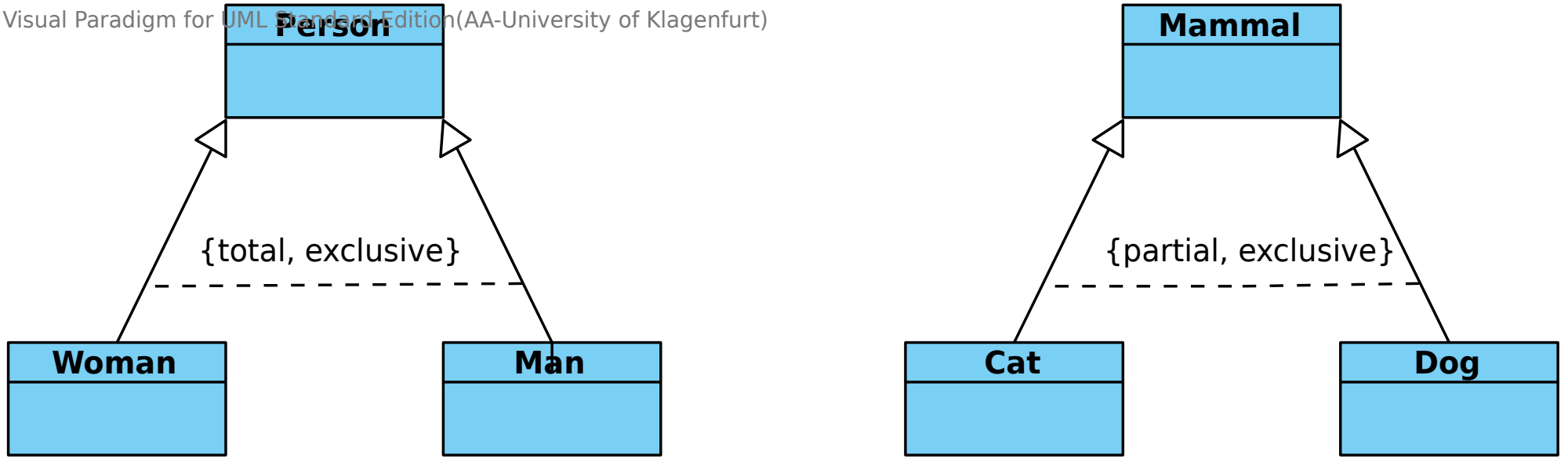
- Conclusion:
  - Costs with redundancy: 405.110
  - Costs without: 40.002.110

- Hence we accept this redundancy to improve our read-performance

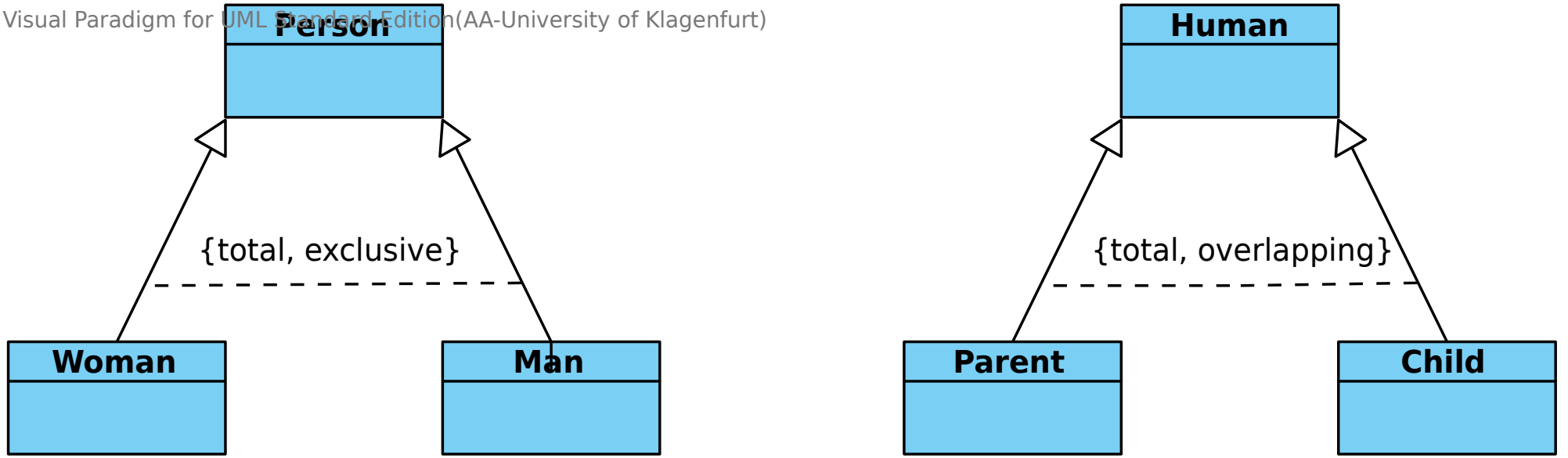
Generalization Hierarchy

- Total vs Partial
- Exclusive vs Overlapping
- Resolving generalization hierarchies
  - generalization not supported by target system
  - transformation from UML to relational model

Total vs Partial



Exclusive vs Overlapping



Flattening Strategies

Ceiling

- every hierarchy

Floor

- total exclusive hierarchies

Cohesion

- every hierarchy

Which strategy to choose

Example

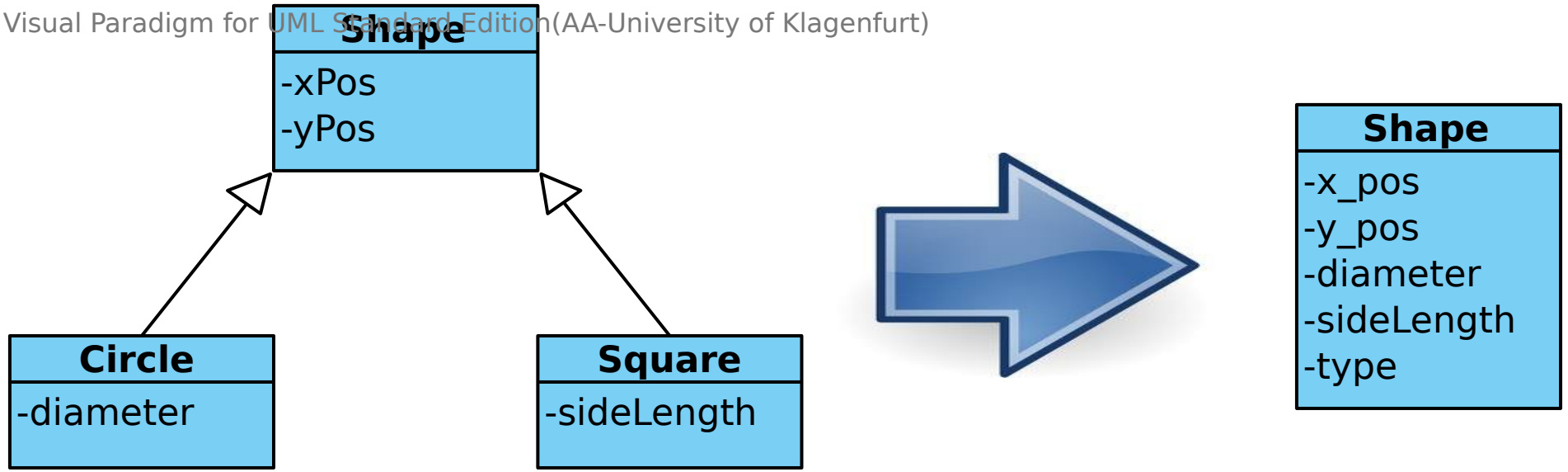
id	given_name	surname	birthday	wage
1	Hans	Hecht	17.02.76	2000
2	Emma	Maier	27.01.61	2300
3	Horst	Adler	11.11.59	2200
4	Michaela	Geiger	23.07.82	3300
5	Christian	Gruber	03.09.73	4000
6	Andrea	Holzhacker	15.05.85	2500
7	Sarah	Steinmetz	12.12.81	1800

Ceiling

- Super class with all attributes of sub-classes
- Remove sub-entities
  - discrimination attribute

- All instances within one class
- Empty attributes
- Semantics for associations between sub-classes?

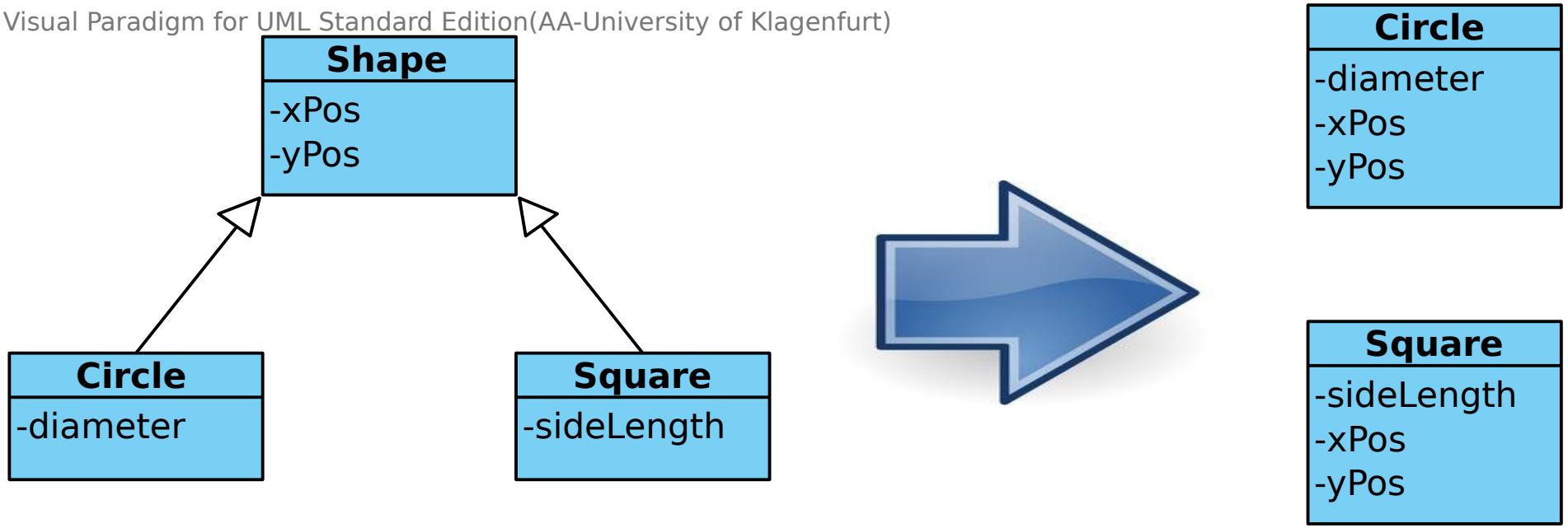
Ceiling


Visual Paradigm for UML Standard Edition(AA-University of Klagenfurt)
Shape
-xPos
-yPos
Circle
-diameter
Square
-sideLength
Shape
-x_pos
-y_pos
-diameter
-sideLength
-type

Floor

- Remove non-leaf classes
  - copy inherited attributes
- Applicable for total and exclusive generalization
- One class per type
- Retrieve all instances of a super type
  - UNION
- Partial hierarchies?
- Overlapping hierarchies?

Floor

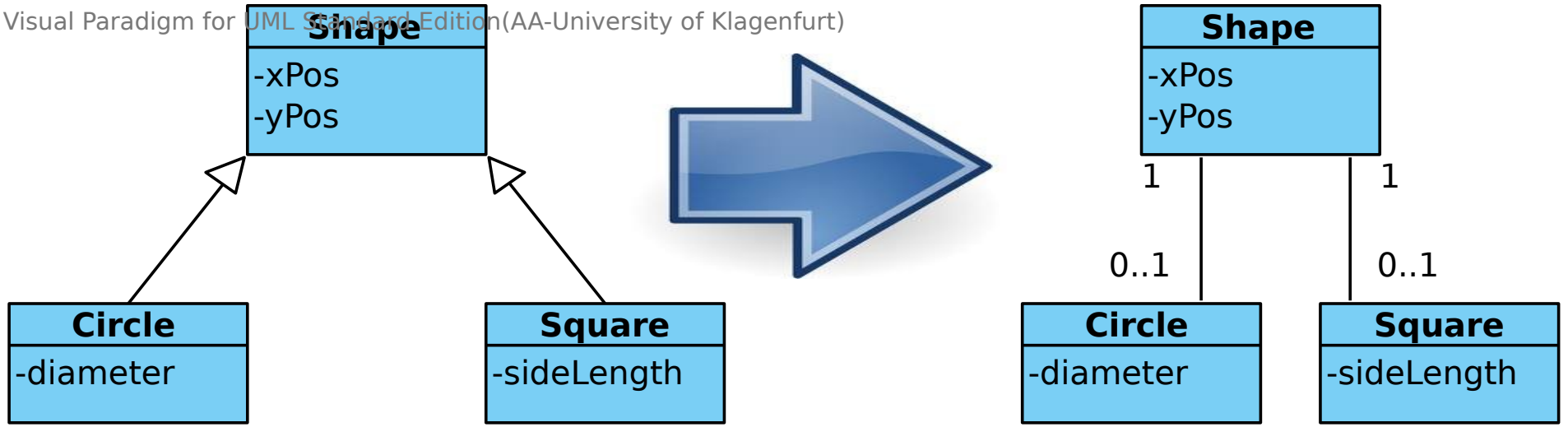

Visual Paradigm for UML Standard Edition(AA-University of Klagenfurt)
Shape
-xPos
-yPos
Circle
-diameter
Square
-sideLength
Circle
-diameter
-xPos
-yPos
Square
-sideLength
-xPos
-yPos

Cohesion

- Generalization represented as 1:1 relationship

- Classes and associations remain
- Retrieve all instances of a super class
  - JOIN
- Loss of generalization semantics

Cohesion


Visual Paradigm for UML Standard Edition(AA-University of Klagenfurt)
Shape
-xPos
-yPos
Circle
-diameter
Square
-sideLength
Shape
-xPos
-yPos
1
1
0..1
0..1
Circle
-diameter
Square
-sideLength

Choosing a flattening strategy

- Identify operation sets
  - super class only (A)
  - super class and exactly one sub class (B)
  - calculate access count

- If A < B: floor

Choosing a flattening strategy

- Otherwise analyze update operations
  - attributes of both super and sub class (C)
  - attributes of either super or sub class (D)

- C > D: ceiling
- Otherwise: cohesion

- Multilevel hierarchies: step-wise bottom up

Partitioning

- Horizontal Partitioning
  - same attribute set
  - UNION
  - multiplication of original associations

- Vertical Partitioning
  - different attribute set
  - JOIN

Example

id	given_name	surname	birthday	wage
1	Hans	Hecht	17.02.76	2000
2	Emma	Maier	27.01.61	2300
3	Horst	Adler	11.11.59	2200
4	Michaela	Geiger	23.07.82	3300
5	Christian	Gruber	03.09.73	4000
6	Andrea	Holzhacker	15.05.85	2500
7	Sarah	Steinmetz	12.12.81	1800

Vertical Partitioning

id	given_name	surname
1	Hans	Hecht
2	Emma	Maier
3	Horst	Adler
4	Michaela	Geiger
5	Christian	Gruber
6	Andrea	Holzhacker
7	Sarah	Steinmetz

birthday	wage
17.02.76	2000
27.01.61	2300
11.11.59	2200
23.07.82	3300
03.09.73	4000
15.05.85	2500
12.12.81	1800

Horizontal Partitioning

id	given_name	surname	birthday	wage
5	Christian	Gruber	03.09.73	1900
2	Emma	Maier	27.01.61	1500
7	Sarah	Steinmetz	12.12.81	1800

id	given_name	surname	birthday	wage
1	Hans	Hecht	17.02.76	2000
6	Andrea	Holzhacker	15.05.85	2500
3	Horst	Adler	11.11.59	2200
4	Michaela	Geiger	23.07.82	3300

Why Partitioning

- Concurrent execution of queries
  - Operations on different sets of instances
  - Operations on different attribute sets
- Reduce network traffic in distributed databases
- Split BLOBs
- Security aspects

Considerations

- Identify bottlenecks
  - reads per day
  - updates per day
  - number of clients
- Query structure
- Index structure
- Enough resources?
- Tuning?