A closer look at the join-equality constraint

Gerhard Skagestein gerhard@ifi.uio.no
Ragnar Normann ragnarn@ifi.uio.no
Department of informatics
University of Oslo

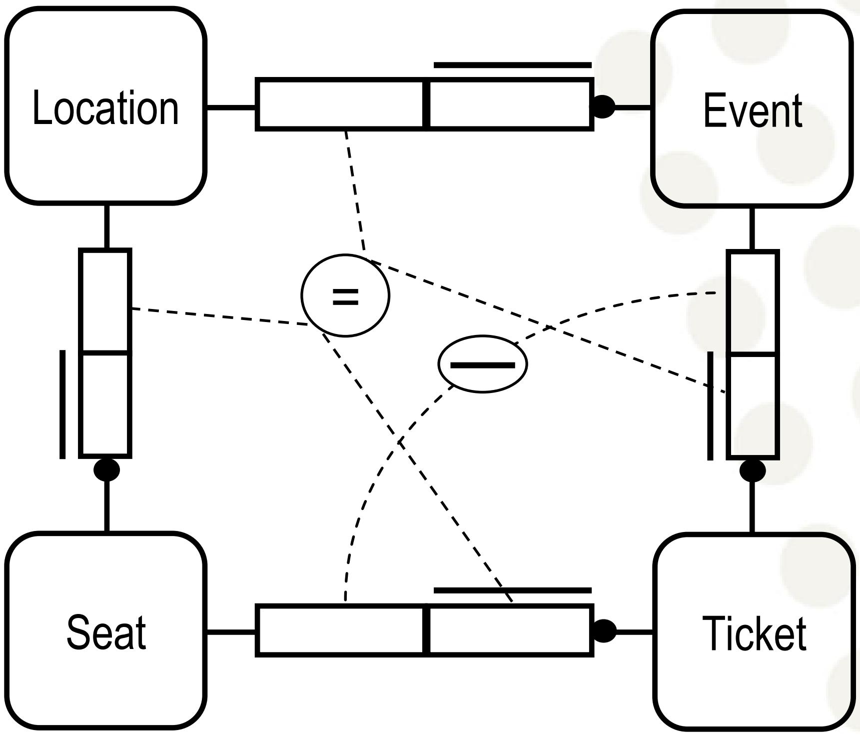
International Workshop on Fact-Oriented Modeling (ORM 2008)
Monterrey, Mexico, November 12-14, 2008

Contents

- The join-equality constraint and
  the history of the ”equivalence of path” constraint.
- An implementation of the join-equality constraint
- The 3NF/BCNF-problem and the join-equality constraint

The join-equality constraint and the history of the ”equivalence of path” constraint.

An example: The ticket model



E

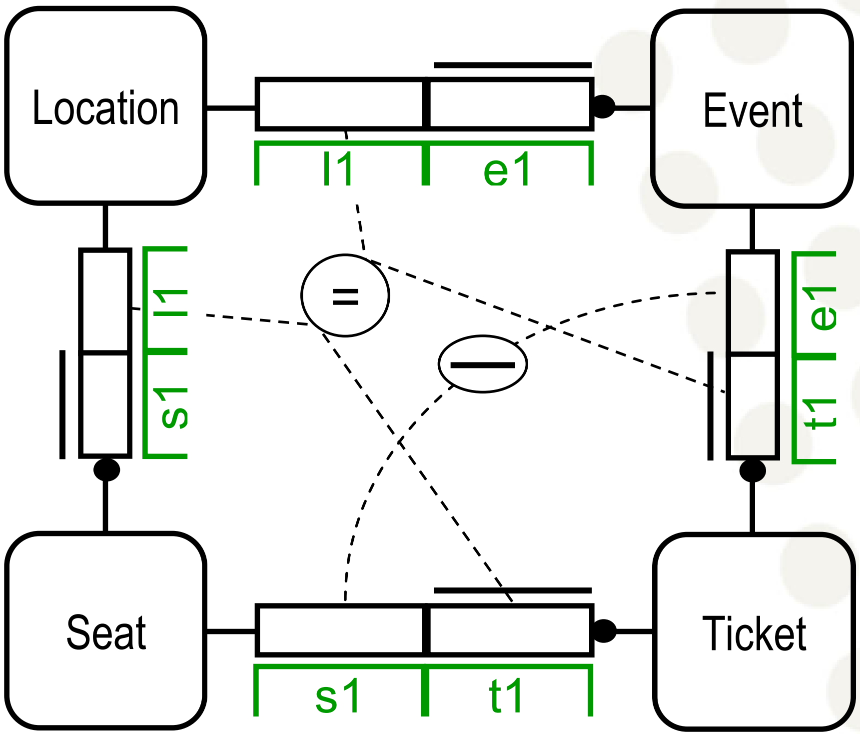
location	event	ticket
l1	e-	t1
l1	e-	t2
~~l1~~	~~e-~~	~~t3~~

S

location	seat	ticket
l1	s-	t1
l1	s-	t2
~~l2~~	~~s~~	~~t3~~

E.ticket = S.ticket ⇒
E.location = S.location

The redundancy can not be removed



The intriguing statement…

- **Halpin, T., Morgan, T.:**
  ***Information Modeling and Relational Databases*, Second Edition, Morgan Kaufmann Publishers, San Francisco (2008)**

  page 405:

The ORM schema in Figure 10.6 includes an equality constraint between role triples, where the first triple involves a join on an objectified association. Role numbers are displayed here to clarify the constraint

How it all began…

- **Spring 1986: Norwegian School of Management, teachers meeting in Ustaoset, Norway:**

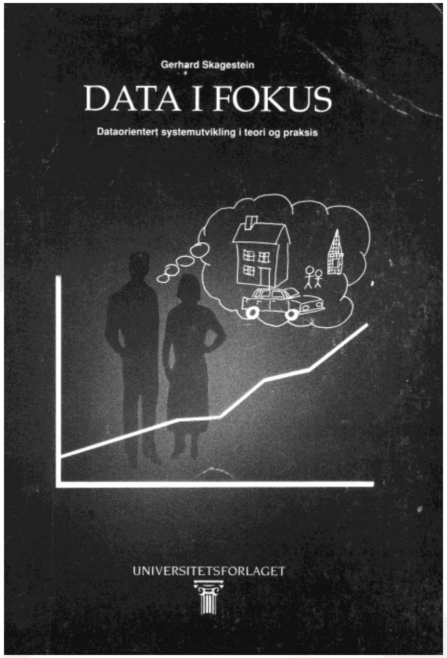
  “The napkin discussion”

- **April 1987 ECODU-43 Davos, Switzerland:**

  The “equivalence of path”-constraint discussed – USA, Netherlands, Norway

History of the equivalence of path constraint

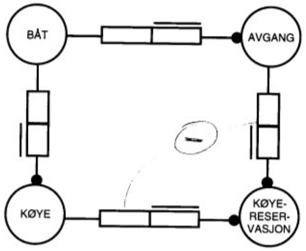
- In the 90ties:
  - No formal graphical notation
  - No international publications?
  - Mentioned in a 1991 Norwegian text-book on data modeling
- Then the theory of join constraints is developed (Halpin 2002)
  - We realize that “equivalence of path” is a special case of the join-equality constraint



«Equivalence of path»-skranken

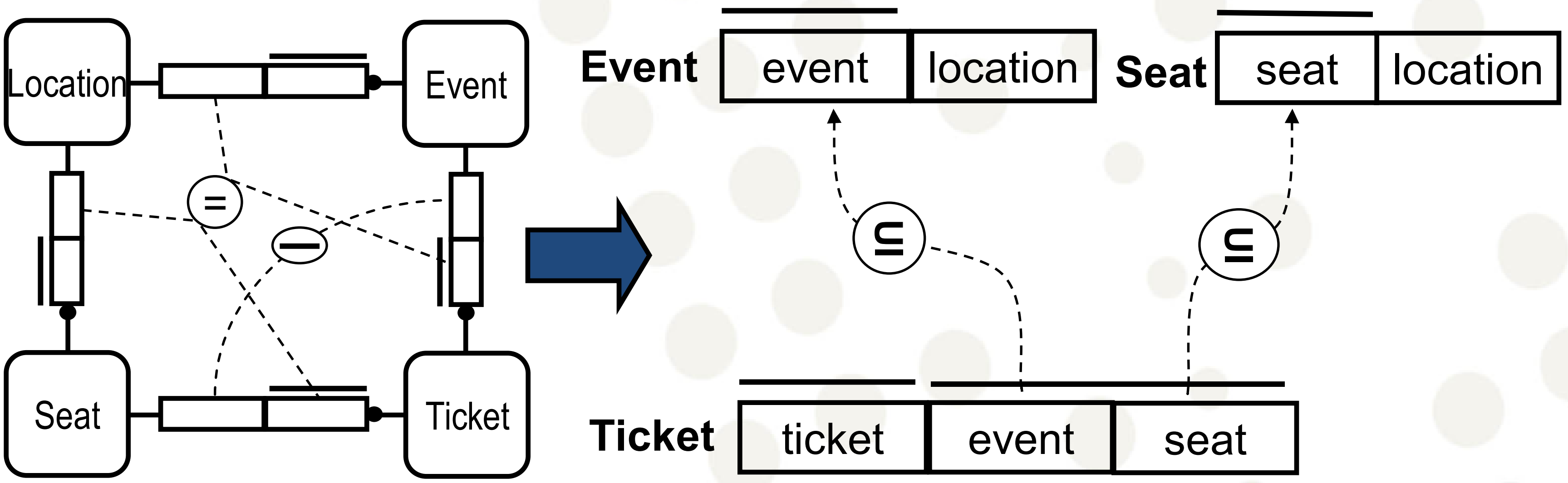
Denne typen skranke forekommer såvidt sjelden og er såvidt avansert at jeg har valgt å ikke oversette betegnelsen på den. «Equivalence of path»-skranken lar seg lettest forklare gjennom et eksempel, se figur 8.18.

Av modellen går det fram at en køyereservasjon gjelder e bestemt køye på en bestemt båt, samtidig som den samm køyereservasjon gjelder en bestemt avgang med en bestemt bå «Equivalence of path»-skranken uttrykker at køya og avgange må gjelde den samme båten! Dessverre finnes det ikke noe.. akseptert standard for å uttrykke «equivalence of path»-skranken grafisk i informasjonsmodellen.



Figur 8.18 Eksempel på «equivalence of path»

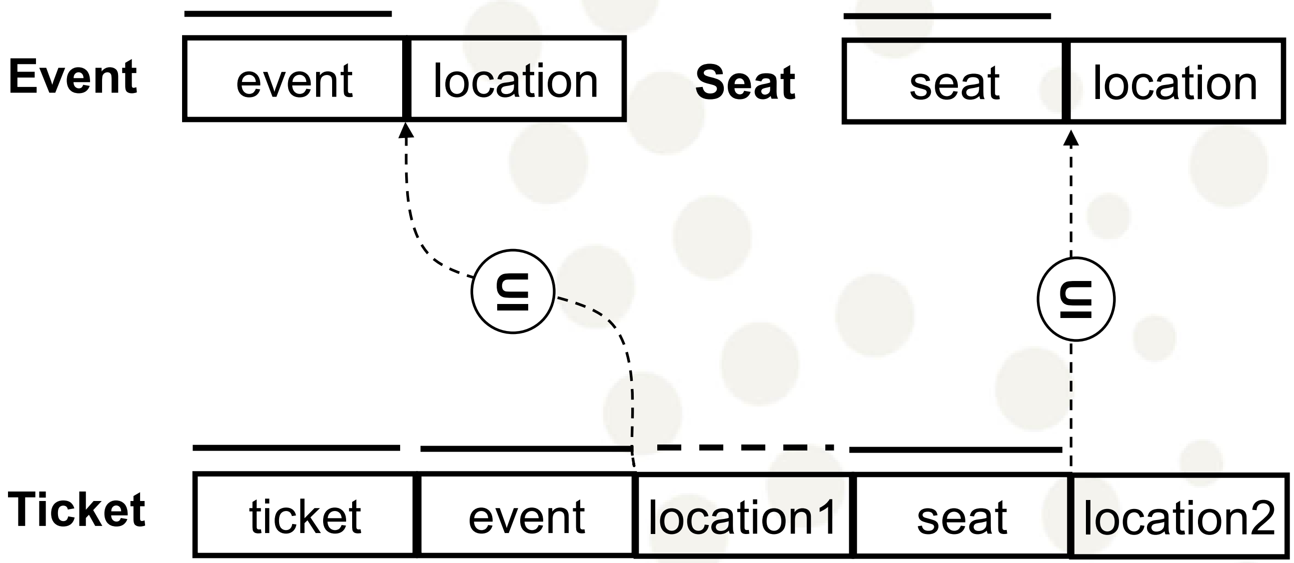
R-map grouping of the ticket model to 3NF



For any **Ticket**,
the **location** of the **Seat**
must be the same
as the **location** of the **Event**.

The ticket model grouped to 1NF

Improved version of Fig. 3 in the paper



For any **Ticket**,
location1 must be the same as **location2.**

The ticket model grouped to 1NF

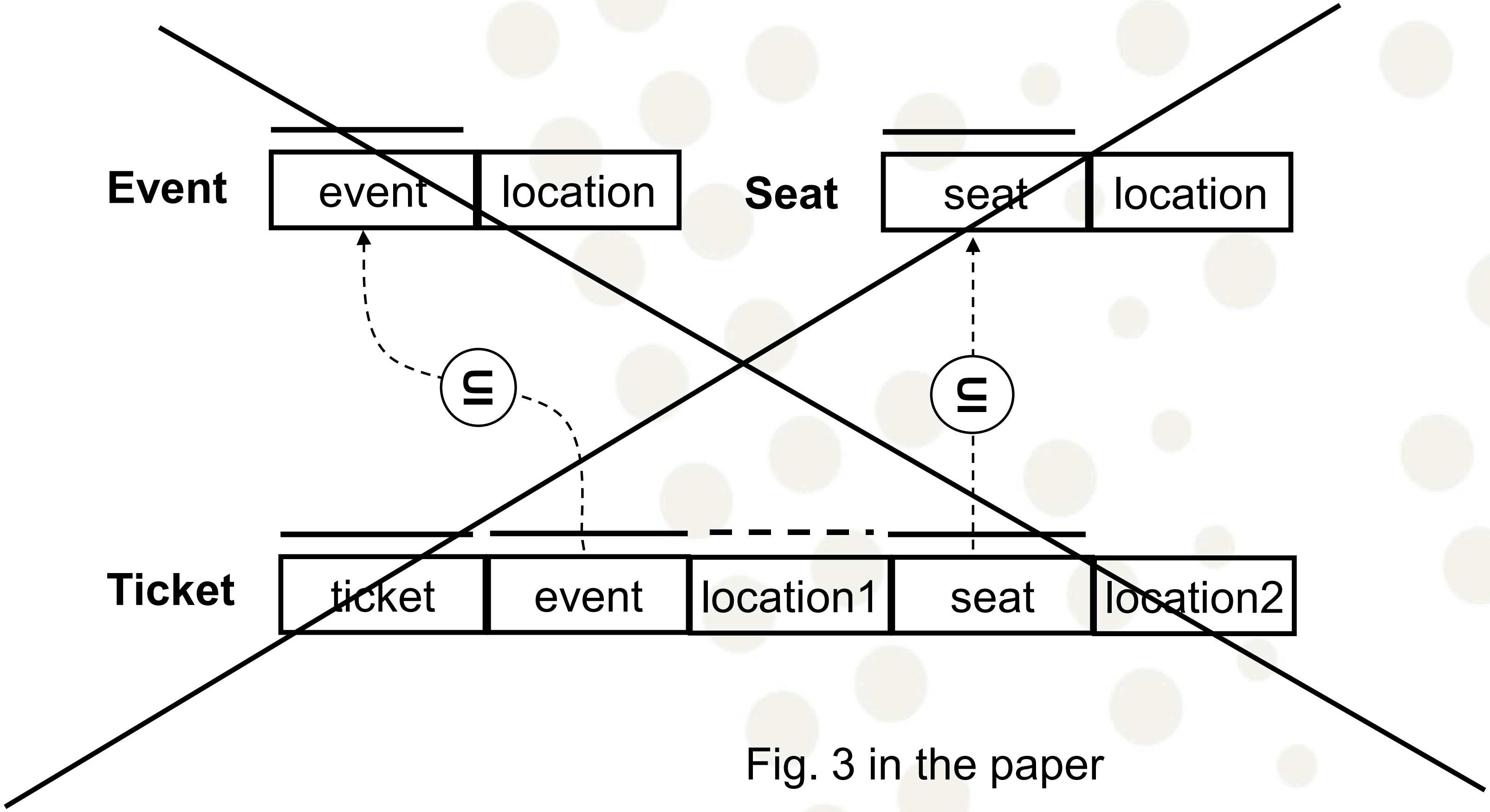
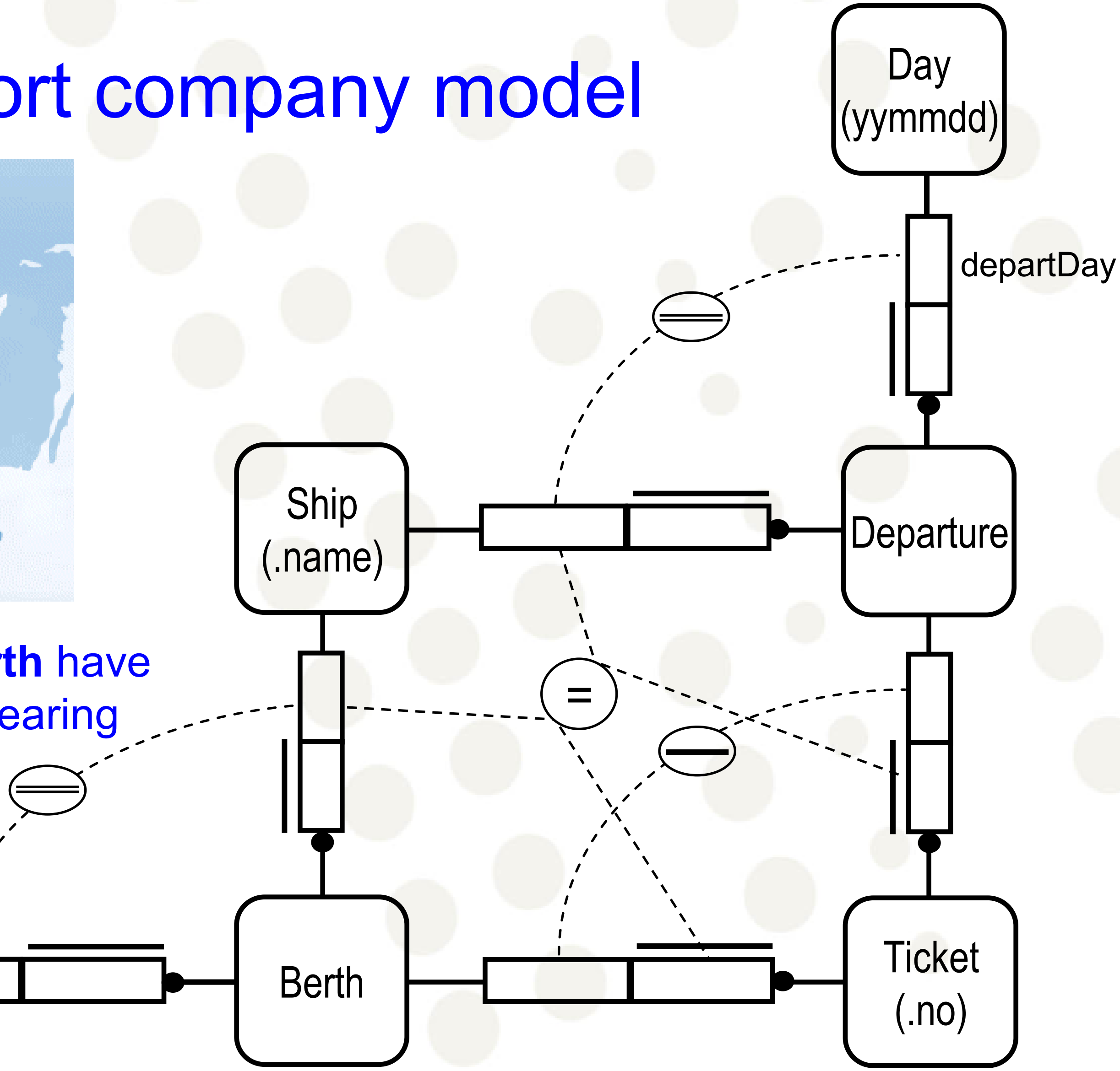


Fig. 3 in the paper

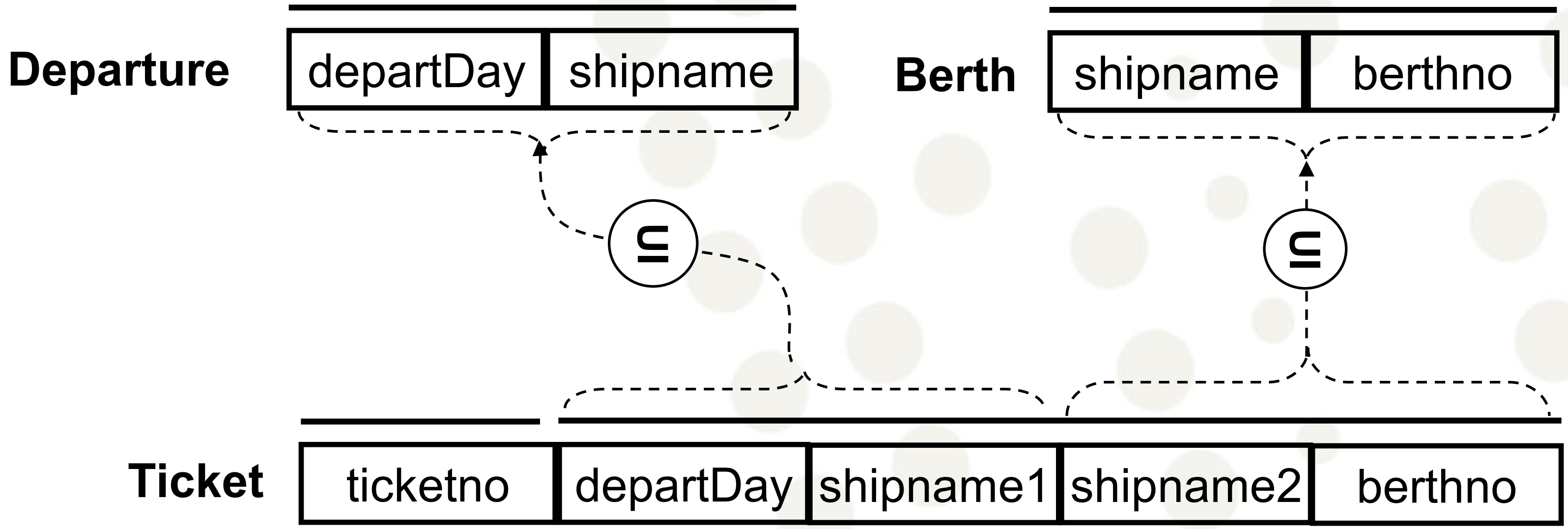
The transport company model

Departure and **Berth** have partly information bearing identifiers**.**



The transport company relational database in 3NF

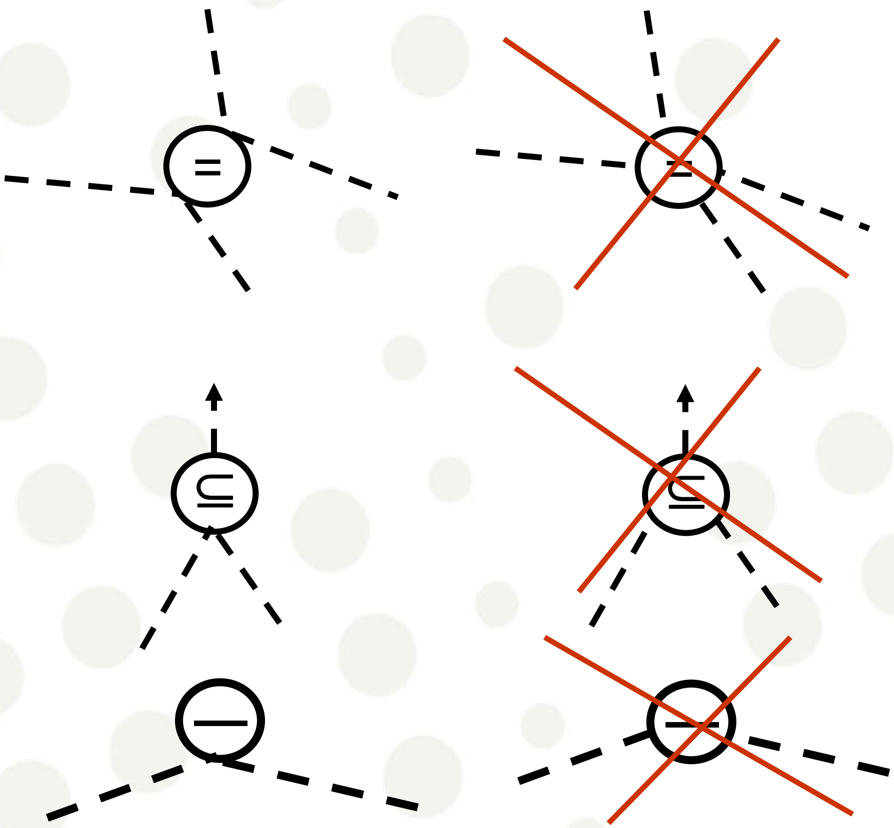
Because of the partly information bearing identifiers of Departure and Berth, the shipnames appear in Ticket even in 3NF.



For any **Ticket**,
shipname1 must be the same as **shipname2.**

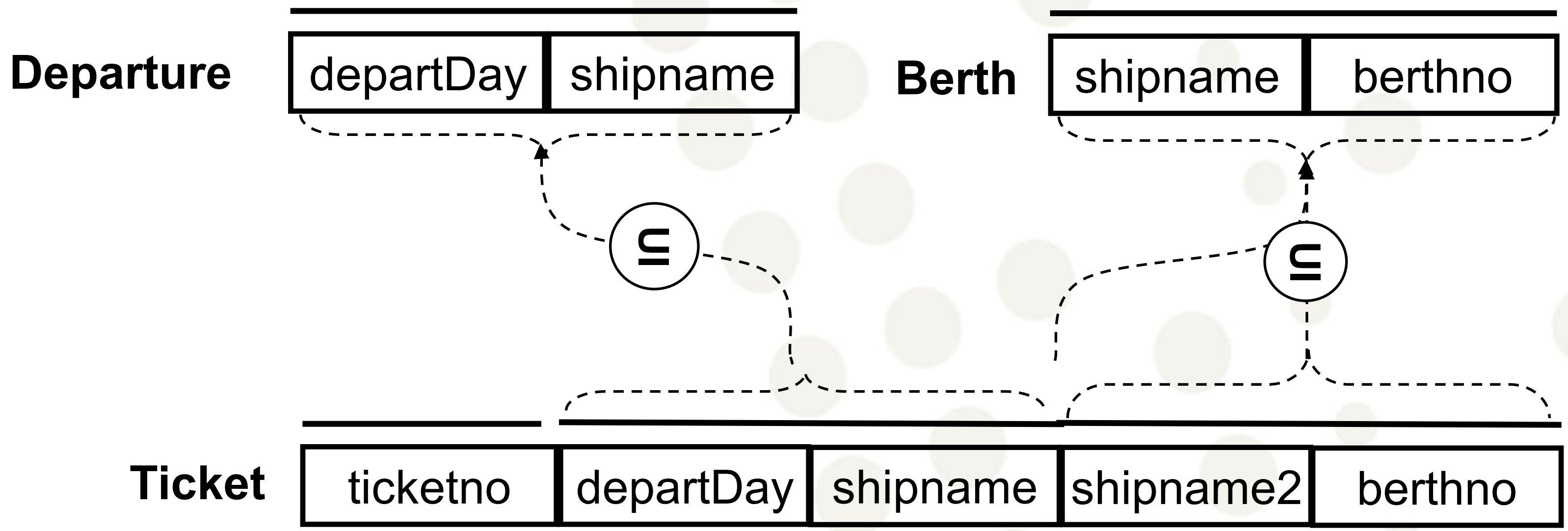
A note on notation

- **In the graphs, we have chosen to indicate the joining by connecting the lines on the outside of the join-constraint symbol, like this:**
- **A join-subset could then look like this:**
- **But as a consequence, join-uniqueness should have been drawn like this:**



An implementation of the join-equality constraint

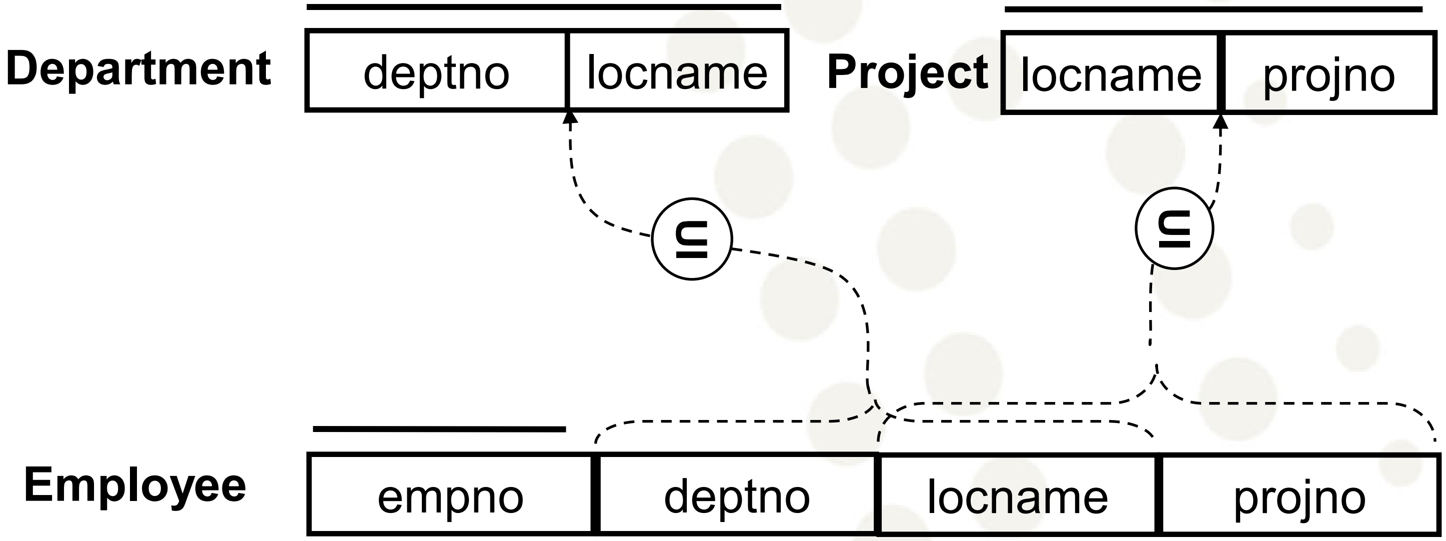
An implementation of the join-equality constraint



Since for all occurrences, **shipname1** is equal to **shipname2**, these two attributes may be replaced by a common attribute **shipname**. This implementation trick gives rise to overlapping foreign keys!

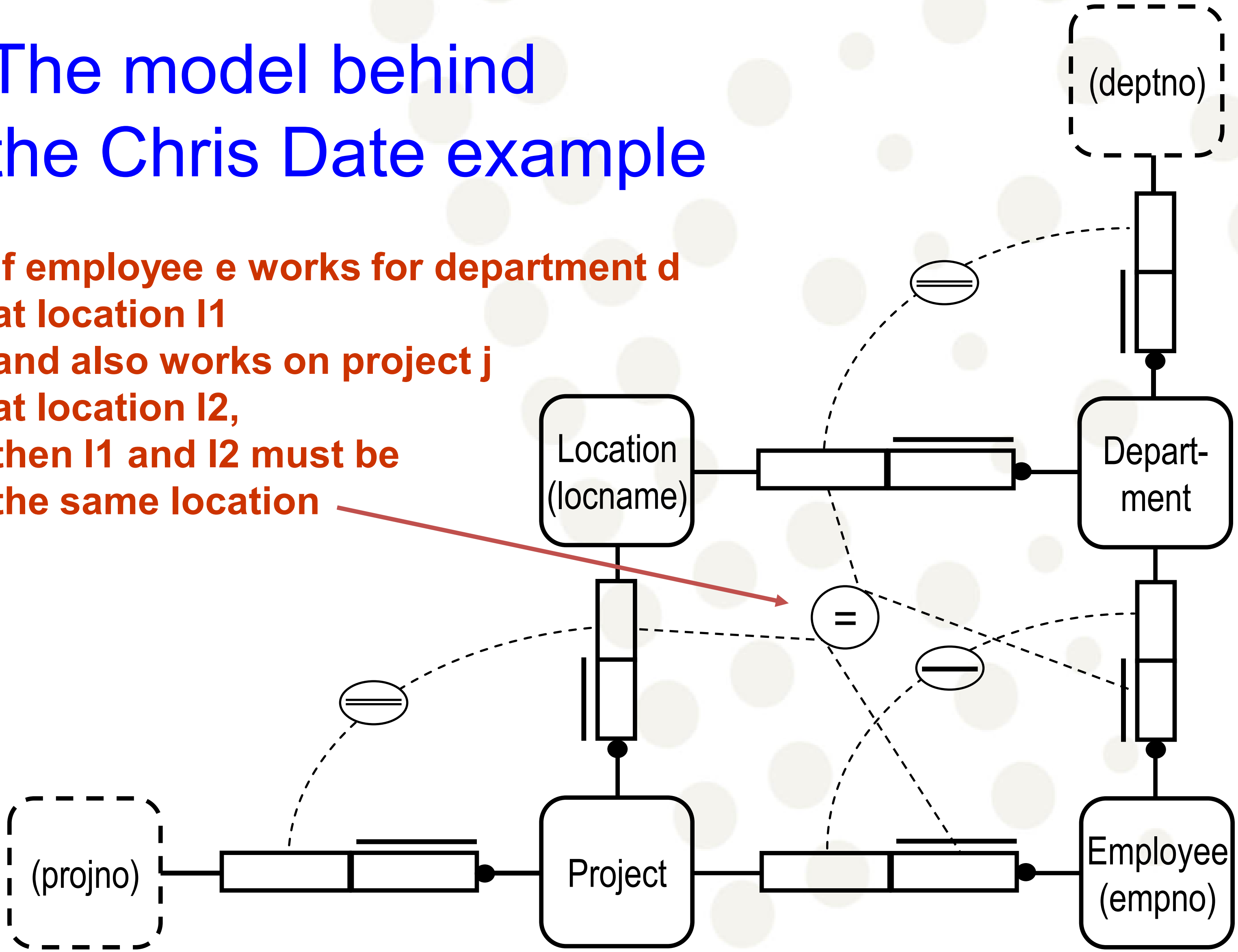
Overlapping foreign keys – a good thing?

Example from Chris Date: Relation Database, Writings 1985-1989, Part I, chapter 18
"Why overlapping keys should be treated with caution“:



The model behind the Chris Date example

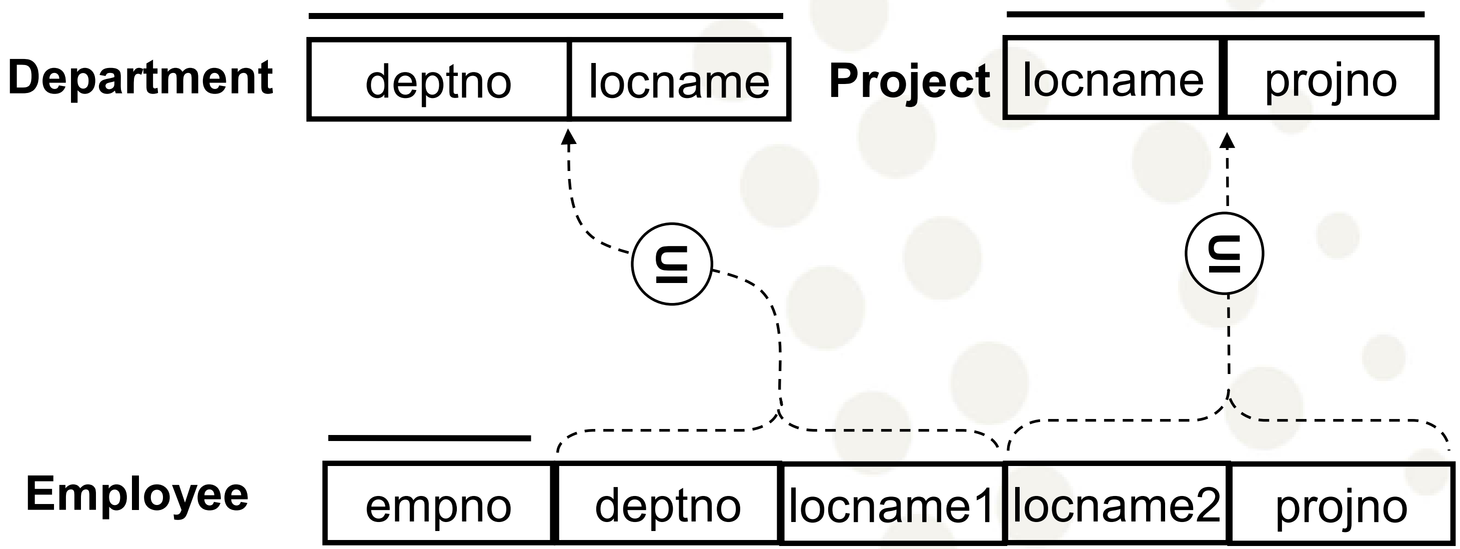
If employee e works for department d at location l1 and also works on project j at location l2, then l1 and l2 must be the same location



A closer look at the join-equality constraint

R-map grouping of the Chris Date model

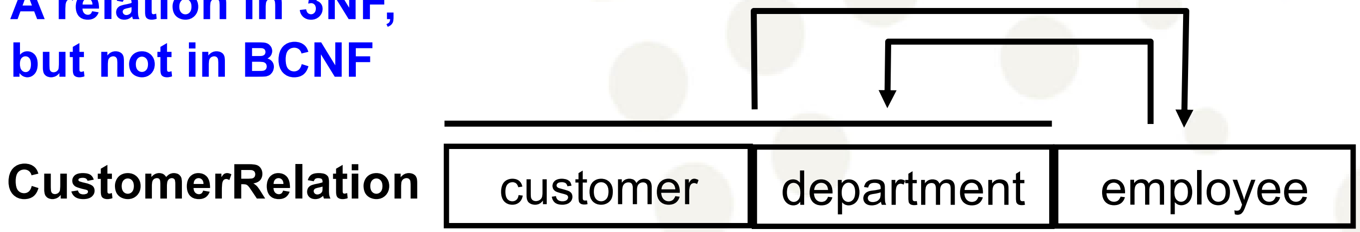
The crucial question:
Is the join-equality constraint really *immutable*,
making it is advisable to implement it by a common attribute?
(I can’t use the term “static constraint” here,
since the ORM community has chosen to use that term for something else...)



The 3NF/BCNF-problem and the join-equality constraint

The 3NF/BCNF-problem

A relation in 3NF,
but not in BCNF

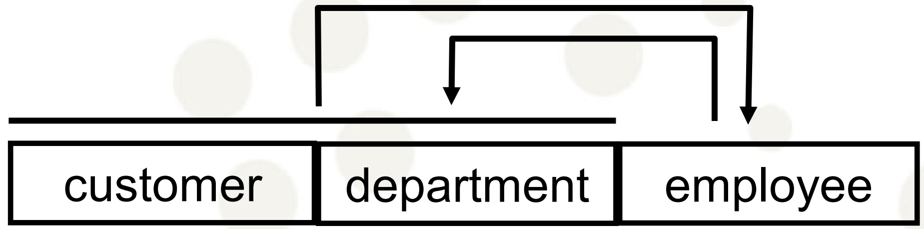


For an employee being responsible for a Customer- Department relationship,
he must work for that Department,
and to work for a department,
the Employee must be responsible for a Customer-Department relationship
for that Department.

The 3NF/BCNF-problem

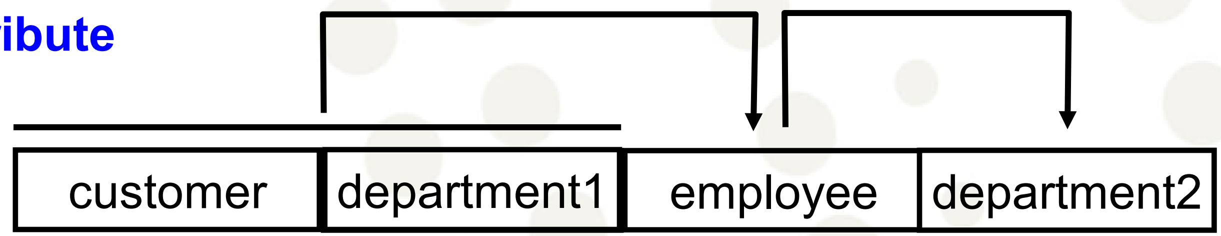
A relation in 3NF, but not in BCNF

CustomerRelation



Converting it to 1NF by duplicating the department attribute

CustomerRelation

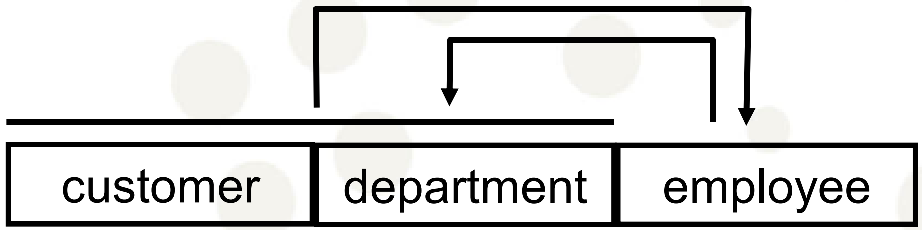


For any **CustomerRelation**,
department1 must be the same as **department2.**

The 3NF/BCNF-problem (animated)

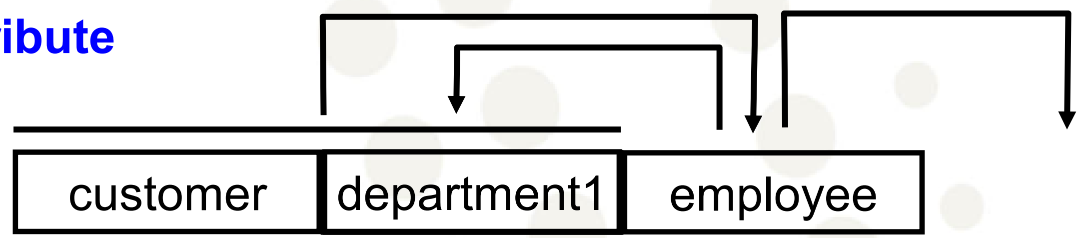
A relation in 3NF, but not in BCNF

CustomerRelation



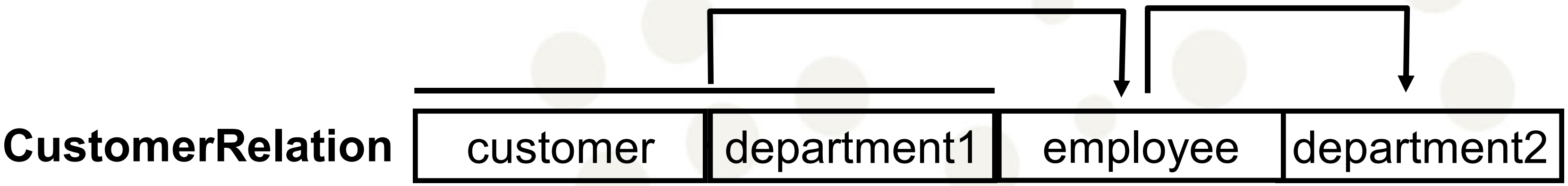
Converting it to 1NF by duplicating the department attribute

CustomerRelation

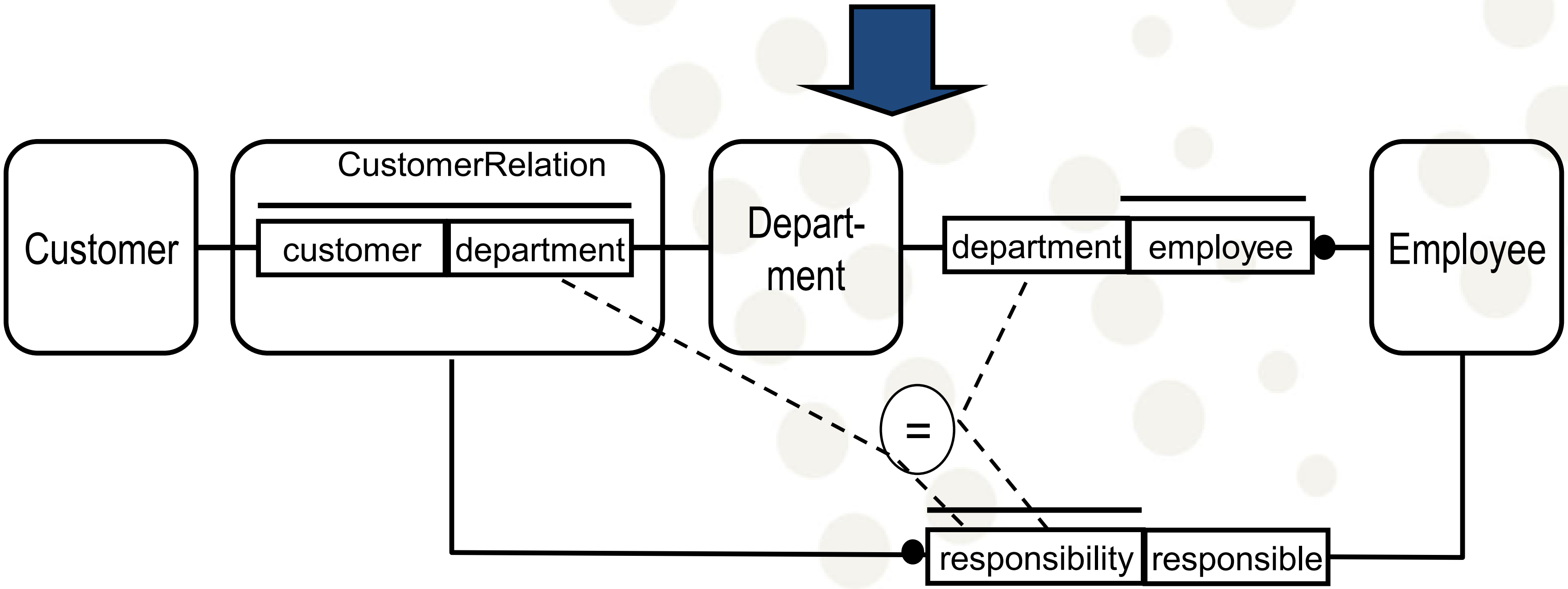


For any **CustomerRelation**,
department1 must be the same as **department2.**

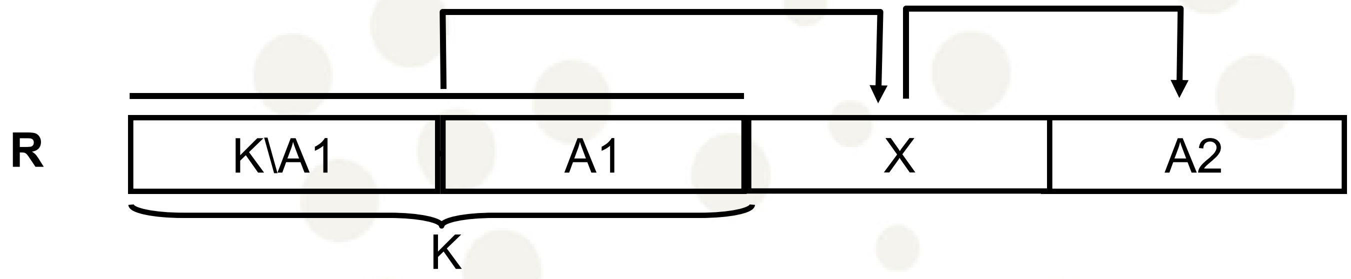
The model behind the 3NF/BCNF-example



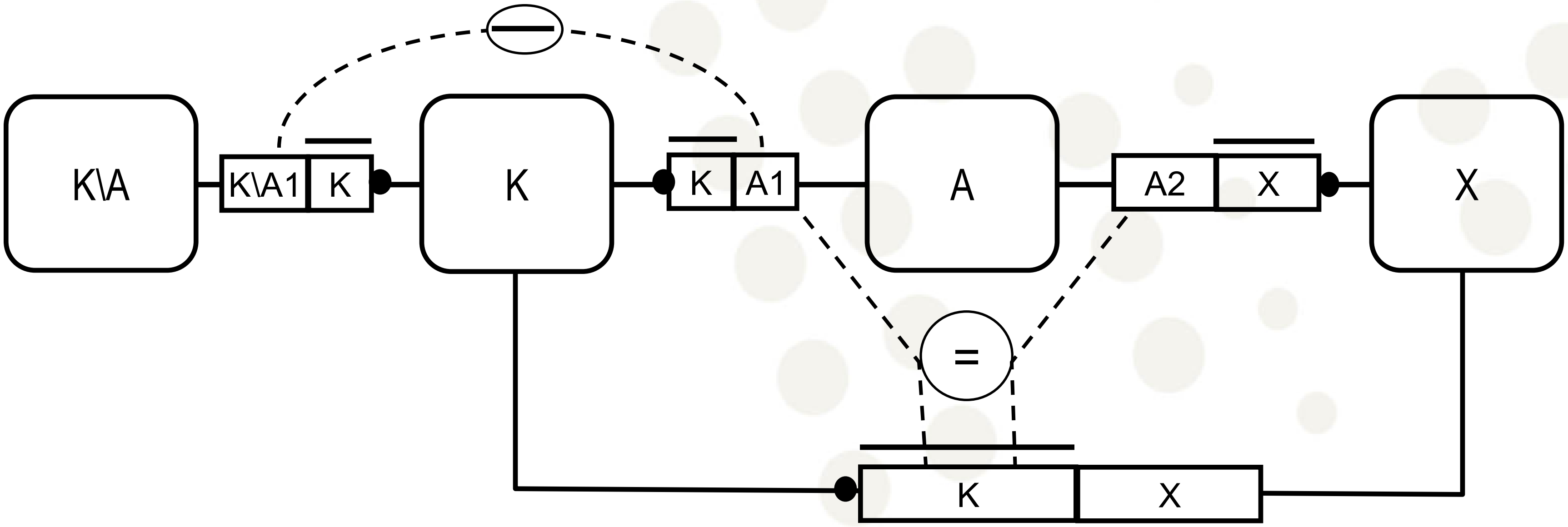
For any **CustomerRelation**,
department1 must be the same as **department2.**



The general 3NF/BCNF-model

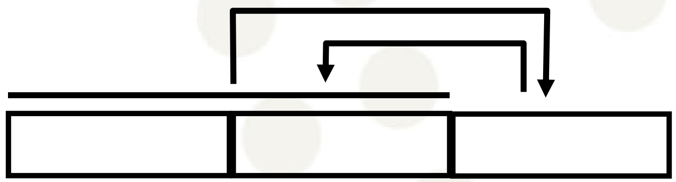


For any K, A1 must be equal to A2.



Observation

- **Whenever we have a relation that is in 3NF, but not in BCNF, there must be a join-equality constraint in the underlying model.**

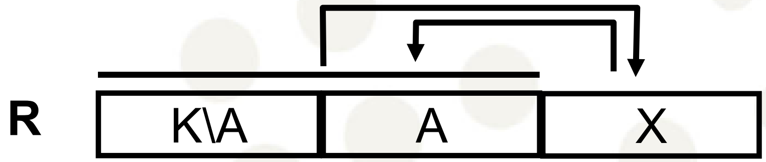


Proof

- **In a relation satisfying BCNF, all non-trivial functional dependencies (FDs) $X \rightarrow A$ must have a superkey as its left hand side X.**

- **In 3NF we in addition allow all FDs $X \rightarrow A$ where A is a key attribute, i.e. $A \in K$ where K is a candidate key.**

Proof (continued)



- **Thus in any 3NF-relation *R* that is not in BCNF, there must be a nontrivial FD $X \rightarrow A$ (i.e. $A \notin X$) where *X* is not a superkey and $A \in K$, a candidate key. Furthermore the FD $K \rightarrow X$ cannot be trivial (this would make $K \setminus A$ a key, violating the minimality of the candidate key *K*).**

- **We then have the trivial FD $K \rightarrow A$ and the two non-trivial FDs $K \rightarrow X \rightarrow A$. In any ORM-diagram having *R* as (part of) its mapped result, these FDs will show up as a join-equality constraint between the two paths from K to A. □**

Summary

- The join-equality constraint *is* rather common.
- The join-equality constraint is inherent in systems encompassing sets of reusable resources subject to reservations, logging or ticketing.
- The join-equality constraint may be implemented by replacing two attributes with one common attribute.
- This implementation may give rise to overlapping foreign keys – overlapping that can be considered safe and sound if the join-equality-constraint is immutable.
- Whenever we have a relation that is in 3NF, but not in BCNF, there must be a join-equality constraint in the underlying model. Hence, a fact-oriented model without any join-equality constraints will group to BCNF when using the R-map procedure.