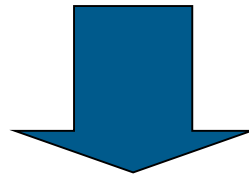


# Generating Relational Model from UML Class Model

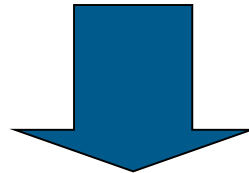
Based on the book by Hovi, Huotari, Lahdenmäki:  
Tietokantojen suunnittelu & indeksointi  
Docendo (2003, 2005) , chapter 7  
© Jouni Huotari & Ari Hovi

# From a conceptual model to physical model

Conceptual model



Logical model: choose a data model



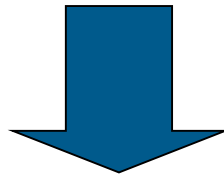
Physical model: select the DBMS

# From a conceptual model to physical model

Defaults in the courses IIO30120 and IIO30220

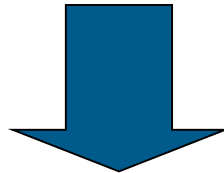
Conceptual model: UML class diagram

Describe the most essential concepts and their relationships



Logical model (database diagram):

Relational model (No many-to-many relationships, 3. NF)



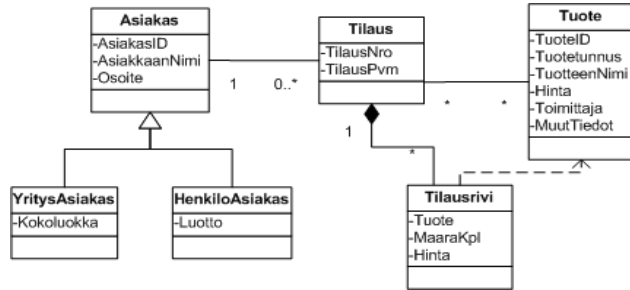
Physical model (create DDL script):

Oracle, SQL Server, MySQL etc.

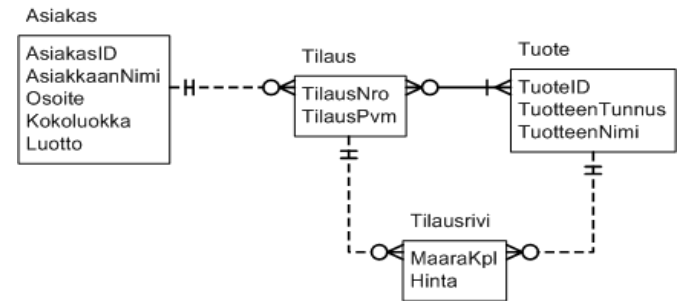
Prototype

# From class diagram to tables

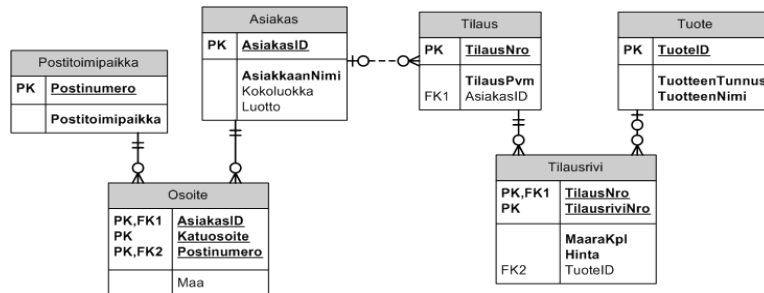
## Class diagram



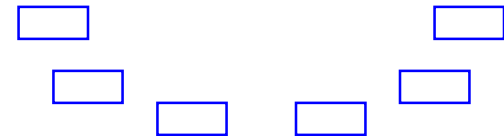
## ER diagram (crow's feet)



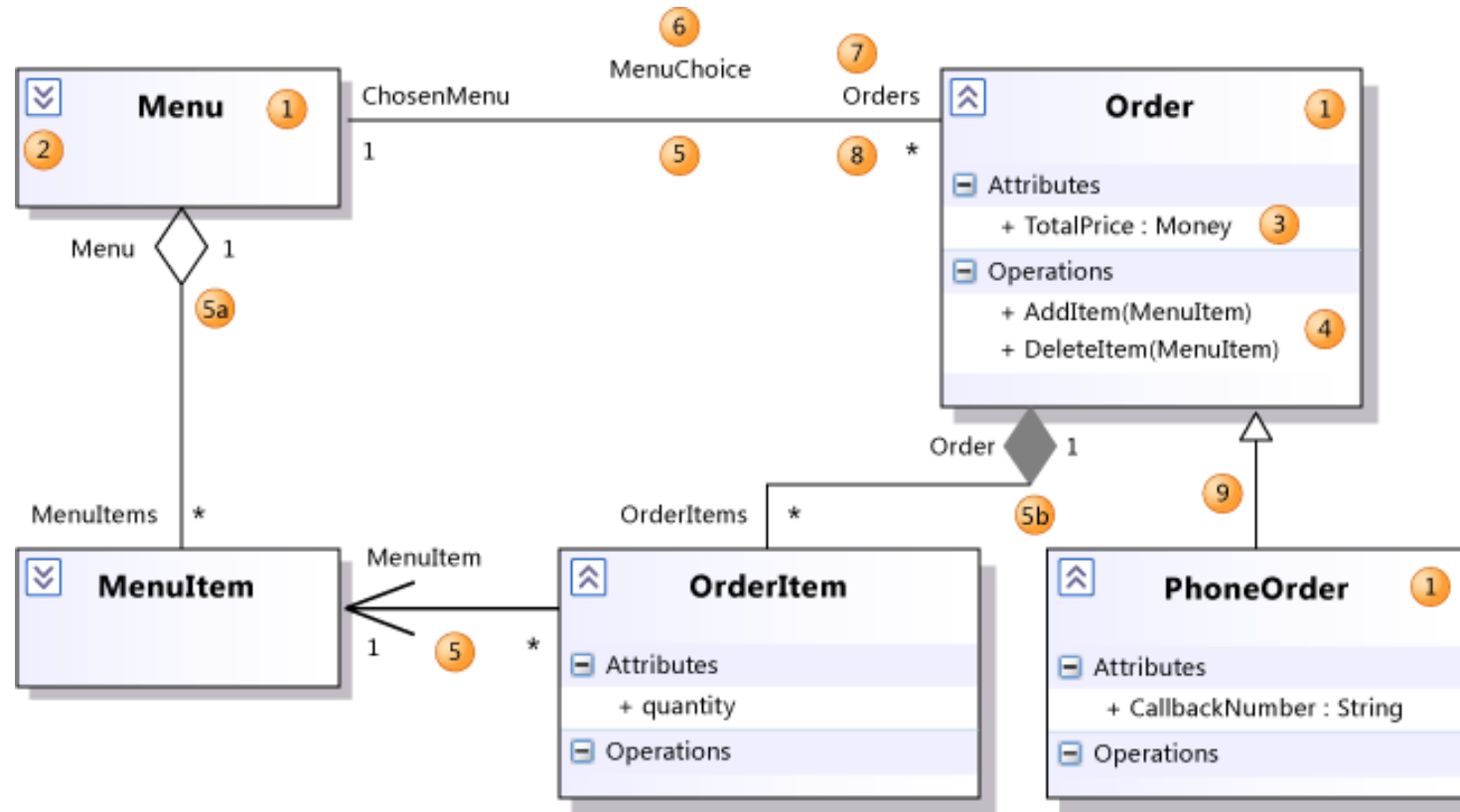
## Table schema (3 NF)



## Database tables



# UML Class Diagrams



<https://msdn.microsoft.com/en-us/library/dd409437.aspx>

<http://www.agilParentdeling.com/artifacts/classDiagram.htm>

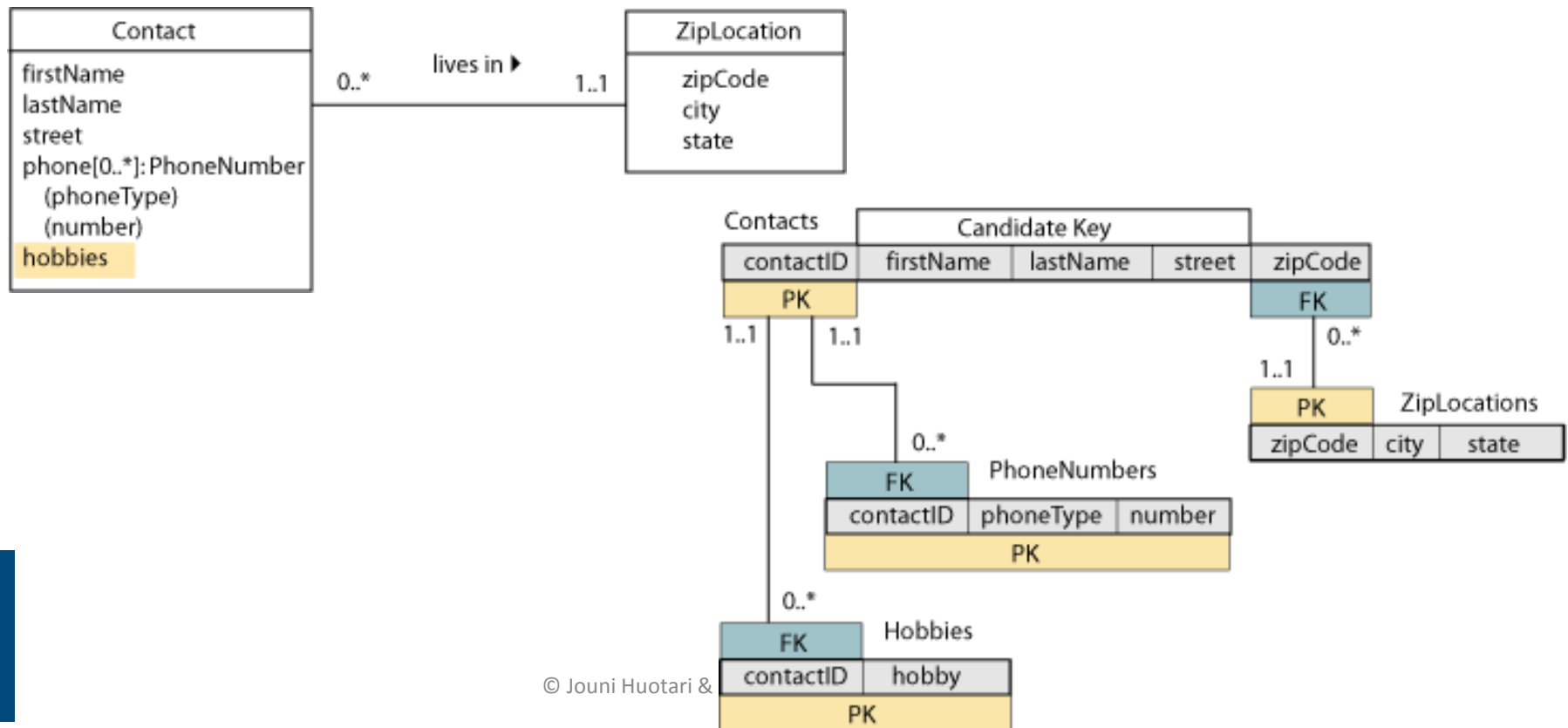
<http://creately.com/diagram-community/popular/t/class-diagram>

# From UML class diagram (conceptual model) to ER diagram (logical model aiming to relational data batabase)

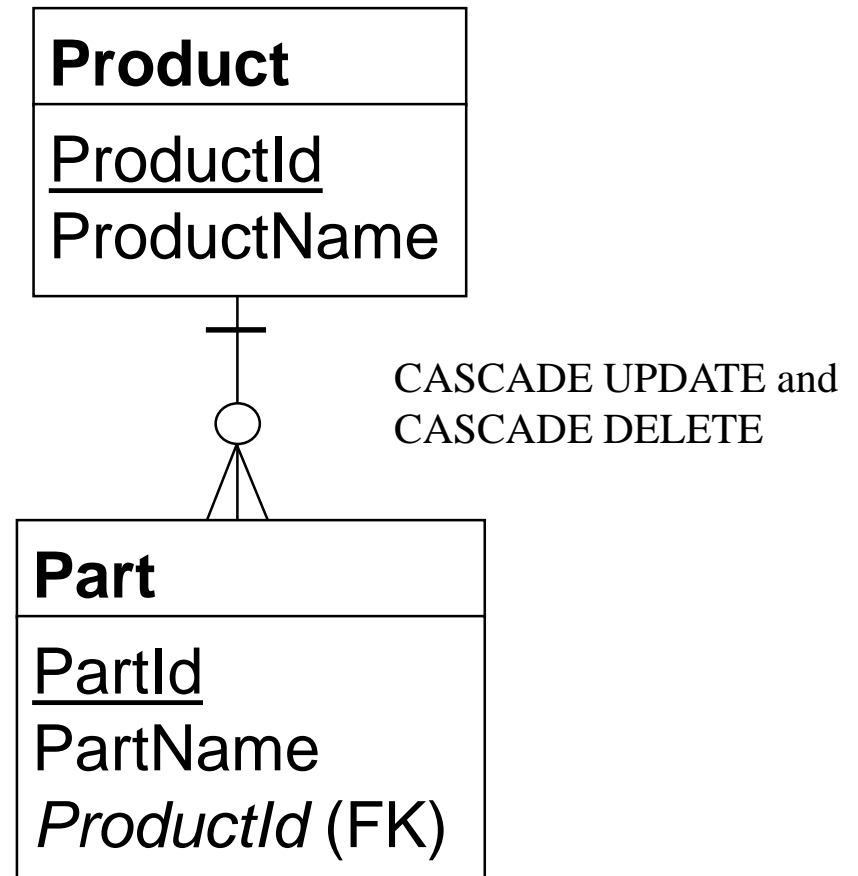
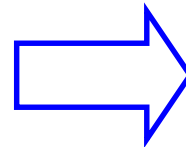
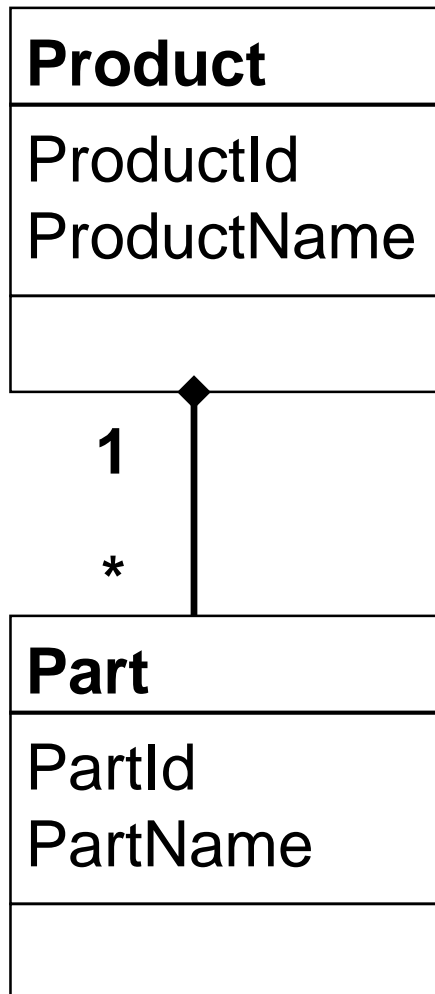
# Multi-valued attribute: create a new table

(cf. 1 NF)

<http://www.tomjewett.com/dbdesign/dbdesign.php?page=hobbies.php>

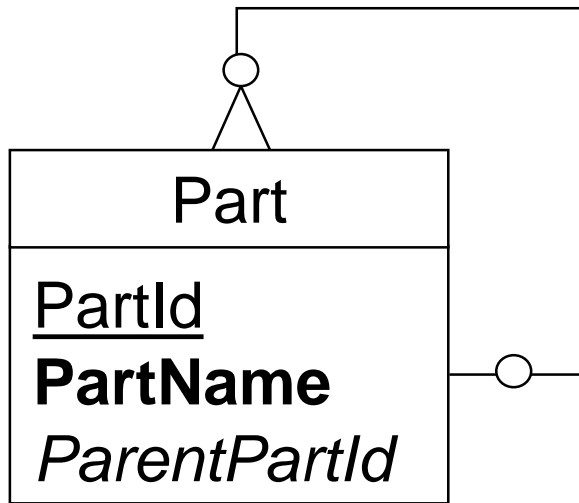


# Aggregation => 1:M relationship

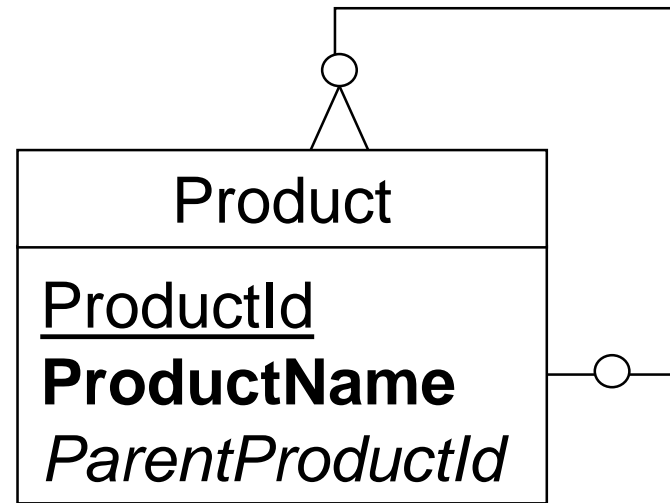




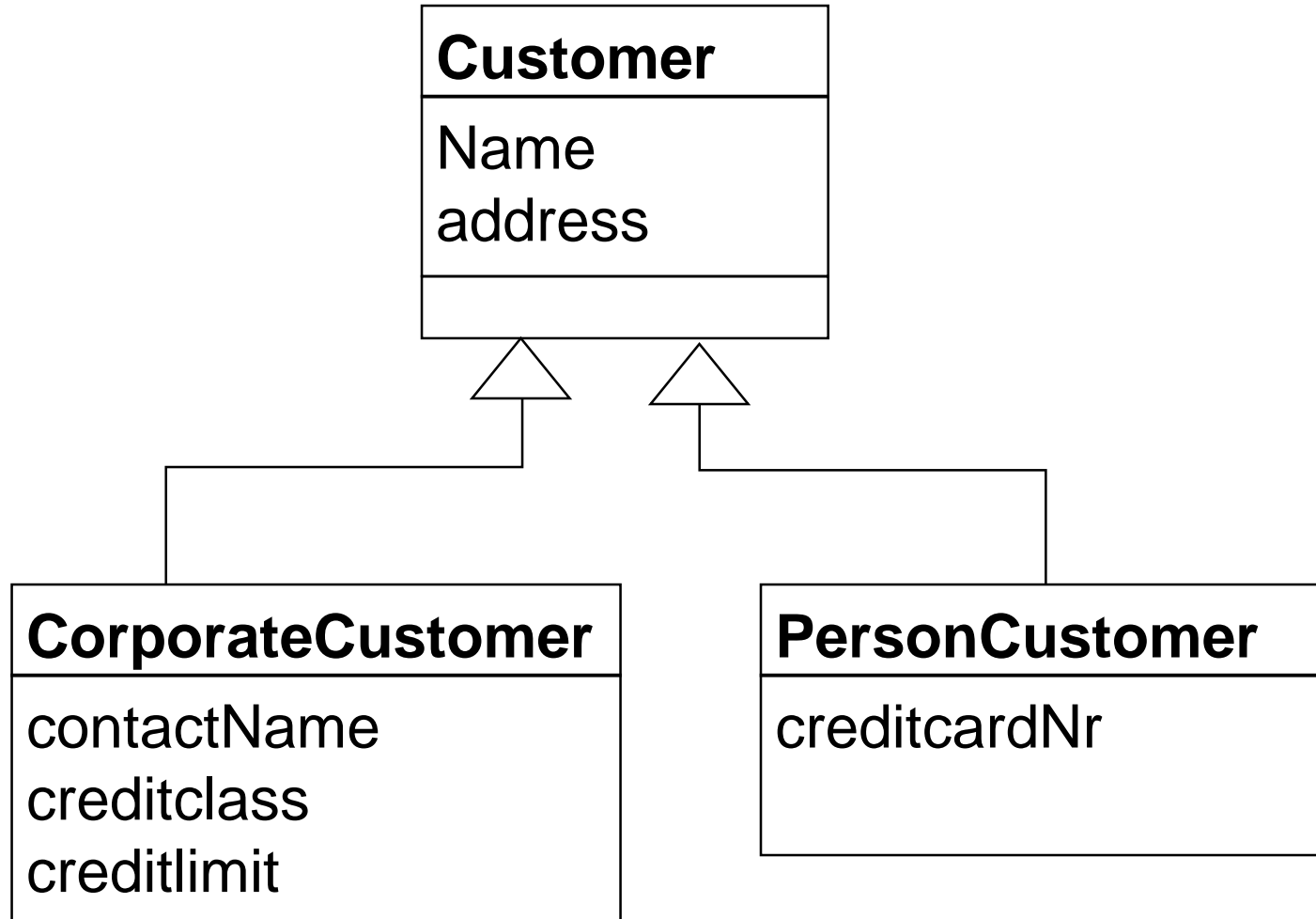
# Aggregation => self-reference relationship



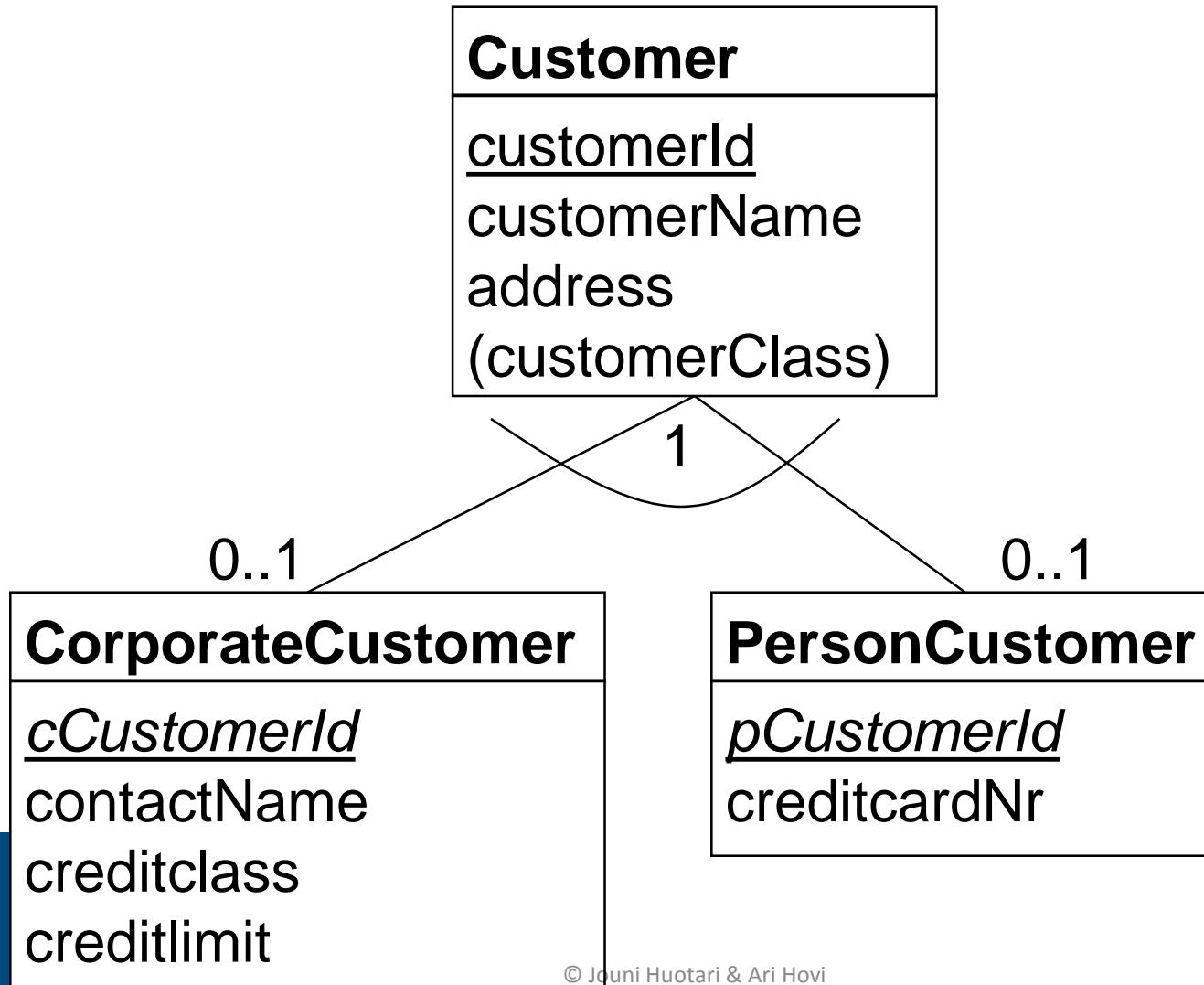
*or*



# Class hierarchy (Fowler 2000)



# Three entities; one for each class



# One entity: combine all classes

<b>Customer</b>
<u>CustomerId</u>
customerName
address
customerClass
contactName
creditClass
creditLimit
creditcardNr

Customerclass: either corporate or person

*Usually the most efficient!*

*Leaves NULLS (but does it really matter?)*

# Two entities: properties of the superclass are placed to the sub classes

## CorporateCustomer

cCustomerId  
customerName  
address  
contactName  
creditclass  
creditlimit

## PersonCustomer

pCustomerId  
personName  
address  
creditcardNr

*Check if there is a relationship between subclasses!*

## Two entities: properties of the superclass are placed to the sub classes

