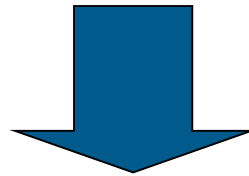


Generating a relational model from a conceptual (UML Class) data model

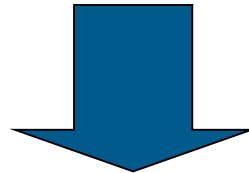


From a conceptual model to physical model

Conceptual model



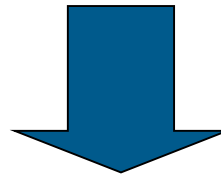
Logical model: choose a data model



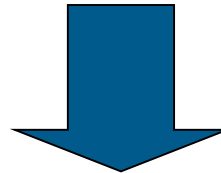
Physical model: select the DBMS

From conceptual data model to physical data model

Conceptual data model: UML class diagram
Describe the most essential concepts and their relationships



Logical data model (database diagram):
Relational model (No many-to-many relationships, 3. NF)



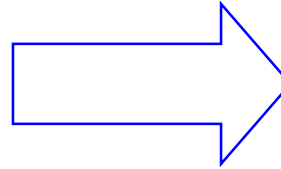
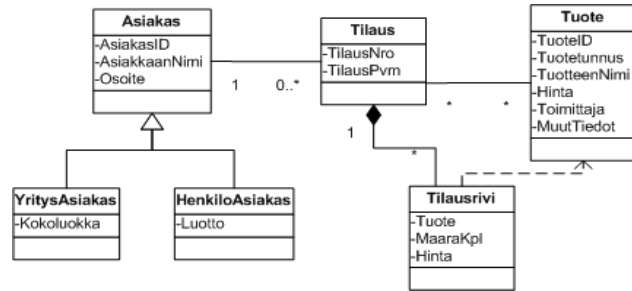
Physical data model (create DDL script):
Oracle, SQL Server, MySQL etc.

Prototype



From class diagram to tables

Class diagram



ER diagram (crow's feet)

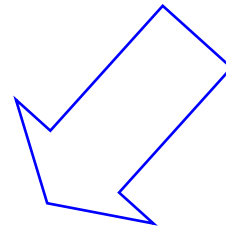
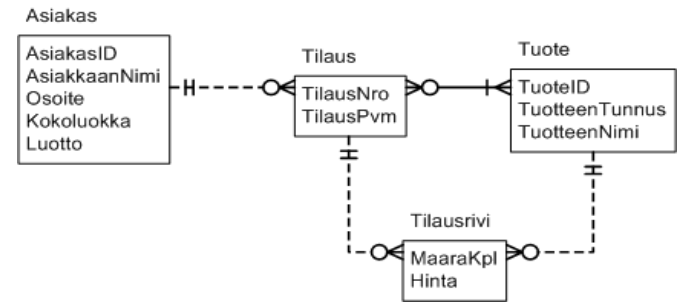
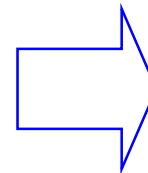
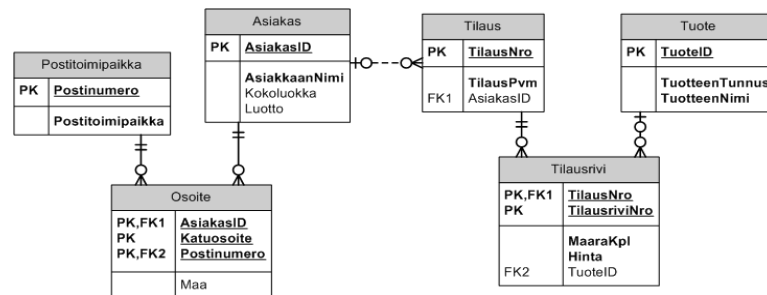
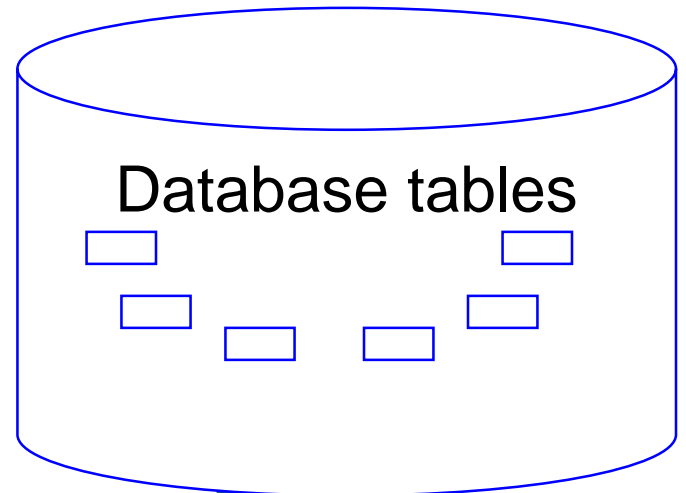


Table schema (3 NF)



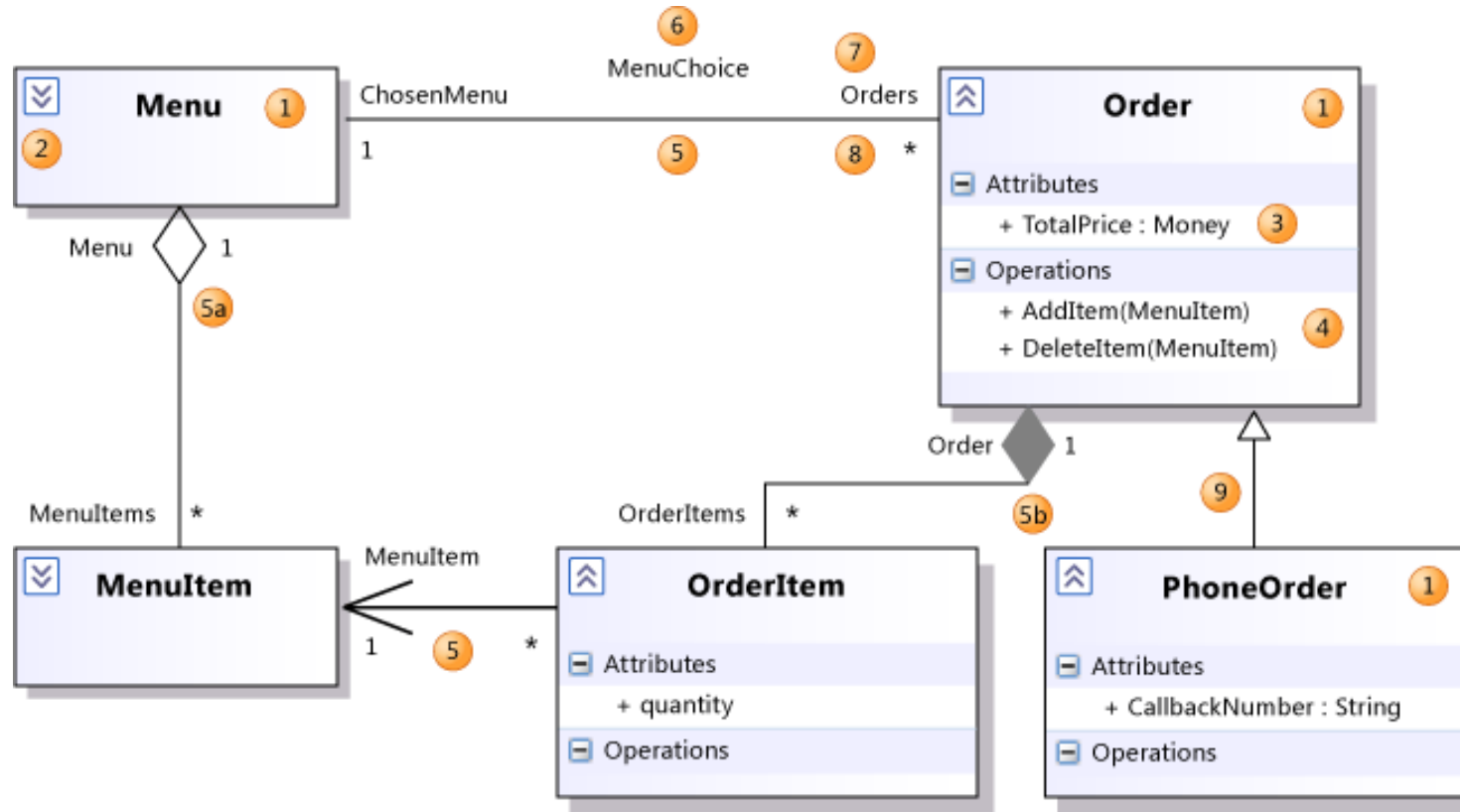
Database tables



https://en.wikipedia.org/wiki/Class_diagram

UML Class Diagrams

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



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

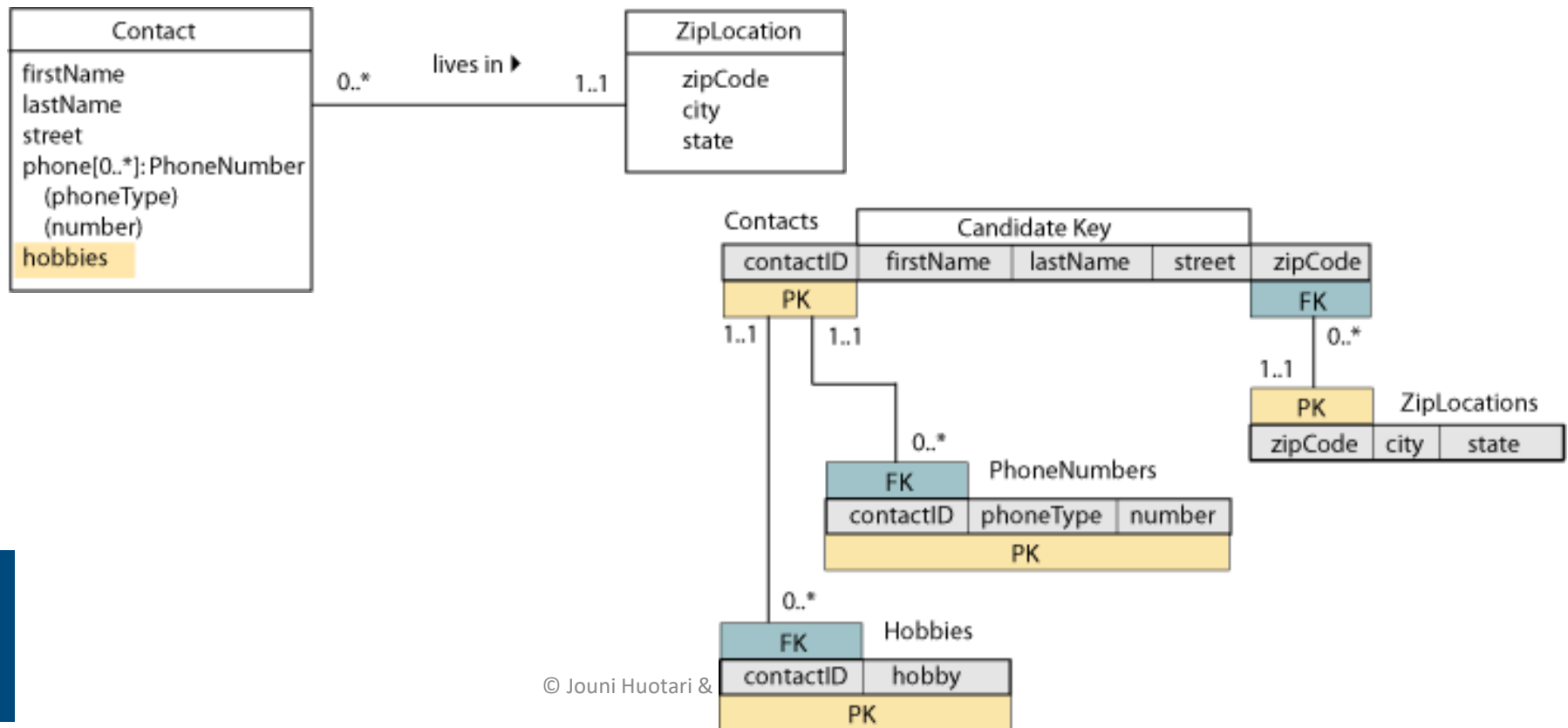
<http://creately.com/diagram-community/popular/t/class-diagram>

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

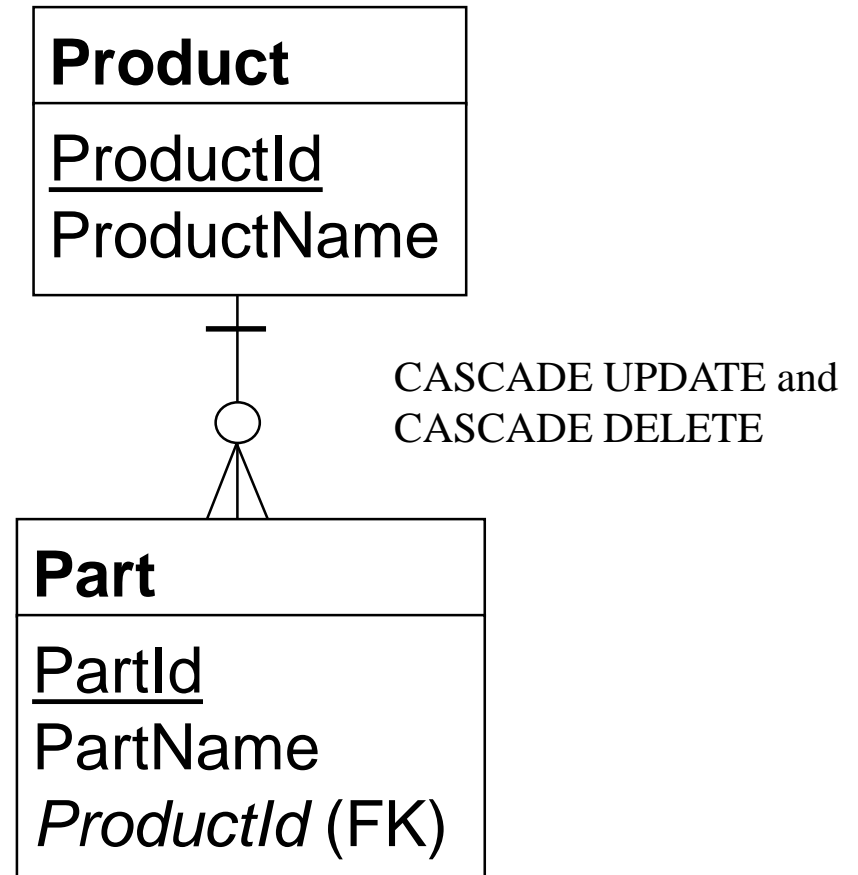
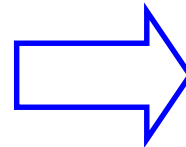
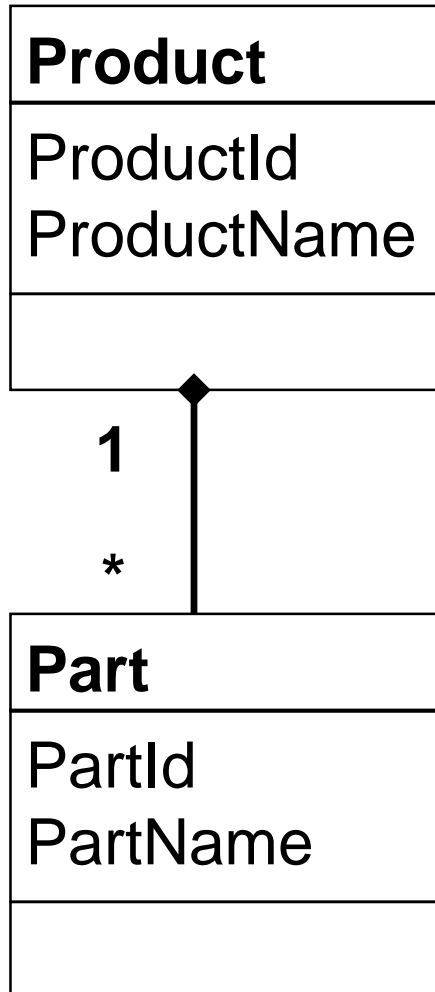
Multi-valued attribute: create a new table

(cf. 1 NF)

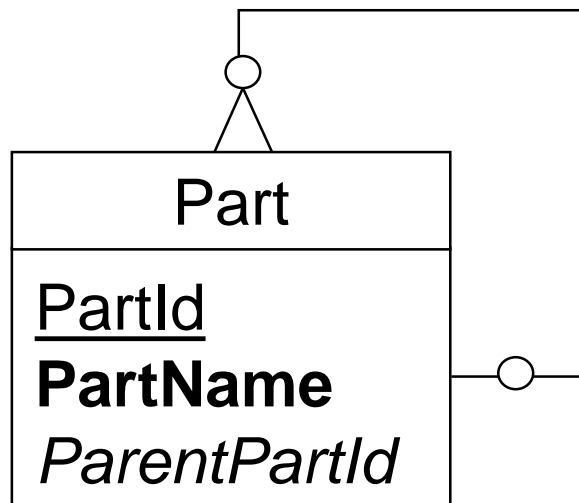
<http://www.tomjewett.com/dbdesign/dbdesign.php?page=hobbies.php>



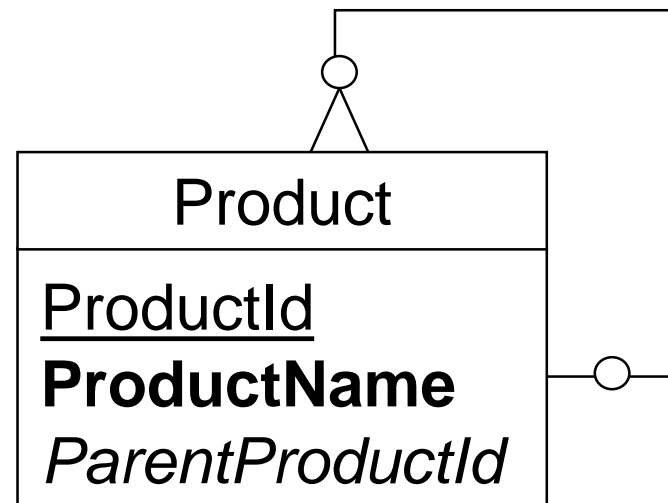
Aggregation => 1:M relationship



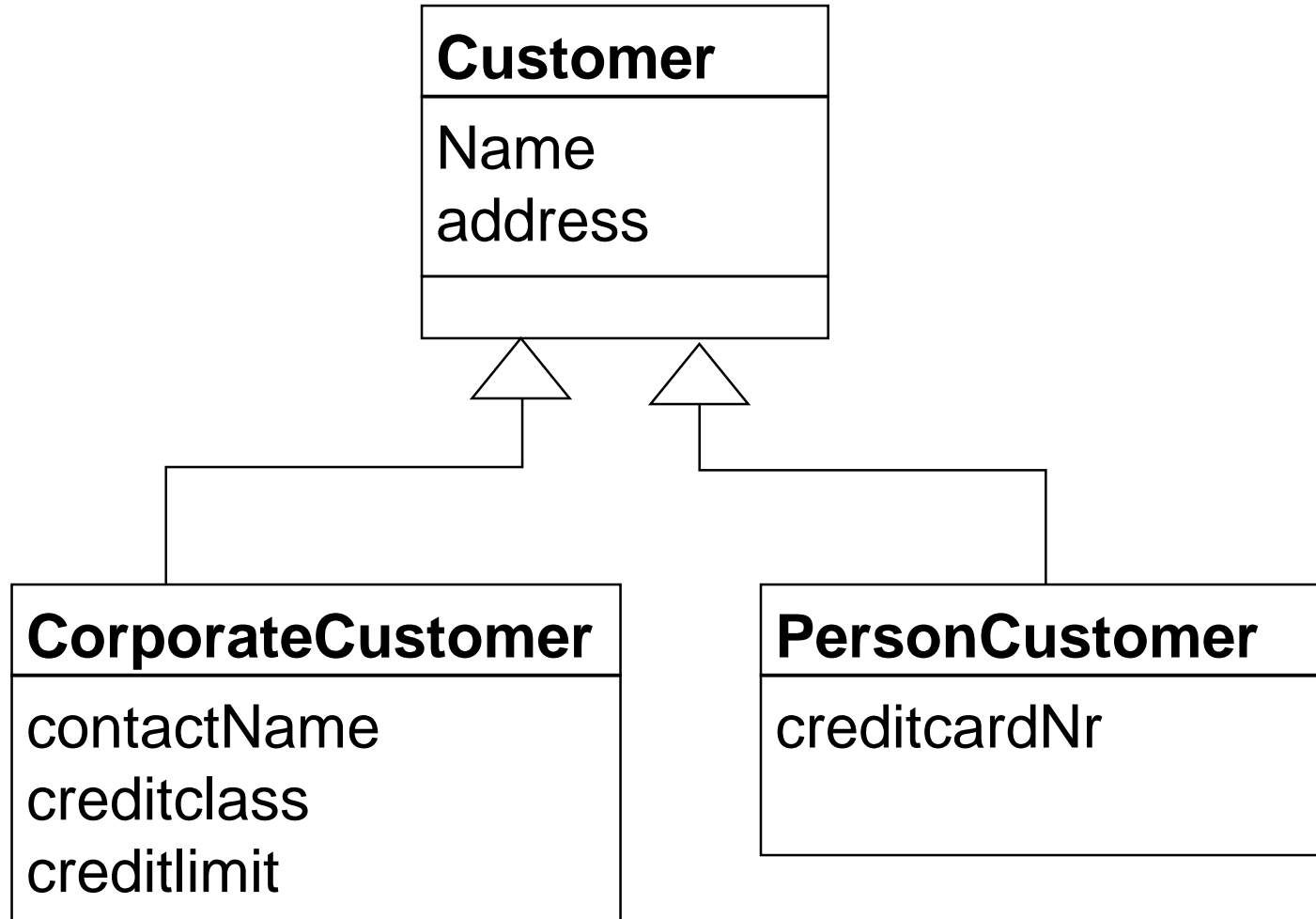
Aggregation => self-reference relationship



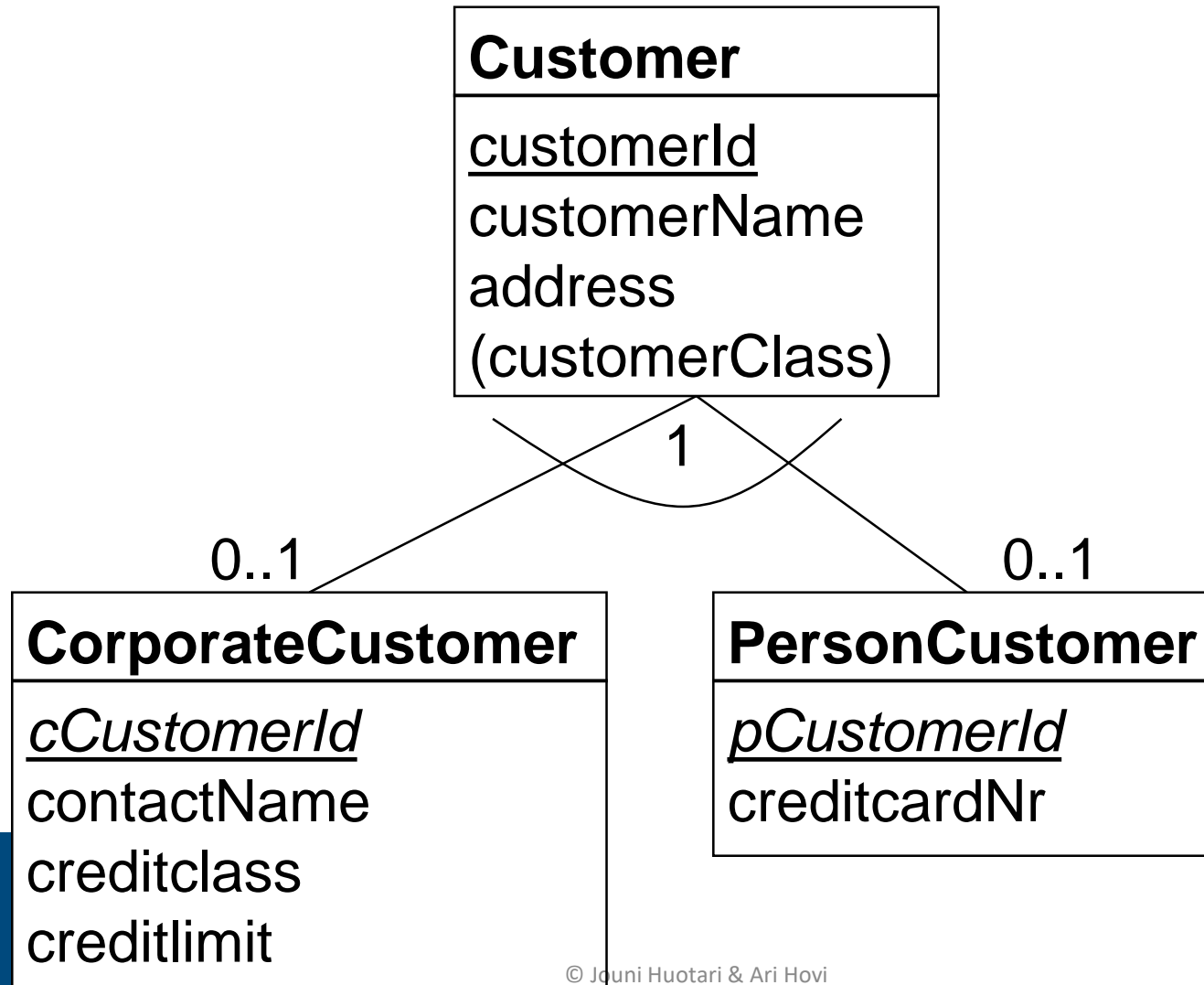
or



Class hierarchy (Fowler 2000)



Three entities; one for each class



One entity: combine all classes

Customer
<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

