Fachhochschule Nordwestschweiz
Hochschule für Technik

Software Architecture

Exercise – ERM versus OOD

BSc

Ingo Arnold

Exercise Opening

Overview

In this example, we consider the address management of a company, in which data of individual persons are stored.

A *person* can assume the following roles: *customer*, *supplier* and *employee*.

All persons share the same attributes: *name*, *first name*, *address* (i.e. street and number), *postal code* and *city*.

Depending on the role of a person, additional attributes are stored. For example, *customers* have an *annual turnover* (cumulative for the current year), *suppliers* have a *quality rating* from previous deliveries, and *employees* work for a *department* and have an *annual salary*.

Exercise Agenda

- ERM versus OOD

ERM versus OOD
Entity Types

The above description gives us a first impression of entities (i.e., entity types) and these entity types’ attributes.

ERM versus OOD
Entity Types

The above description gives us a first impression of entities (i.e., entity types) and these entity types' attributes.

Entity (Entity Type)	Attribute(s)
Person	Name FirstName Address PostalCode City
Customer (role)	AnnualTurnover
Supplier (role)	QualityRating
Employee (role)	Department AnnualSalary

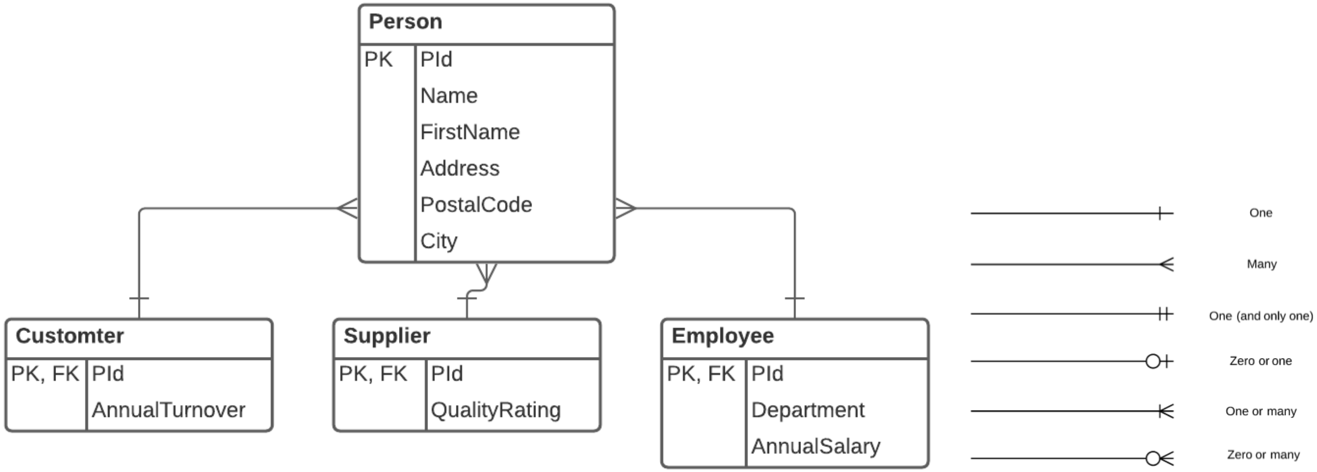
ERM versus OOD
RDBMS Representation

How are these entity types represented in a relational database in the development view?

ERM versus OOD

RDBMS Representation

How are these entity types represented in a relational database in the development view?



Person	
PK	PId
	Name
	FirstName
	Address
	PostalCode
	City

Customter	
PK, FK	PId
	AnnualTurnover

Supplier	
PK, FK	PId
	QualityRating

Employee	
PK, FK	PId
	Department
	AnnualSalary

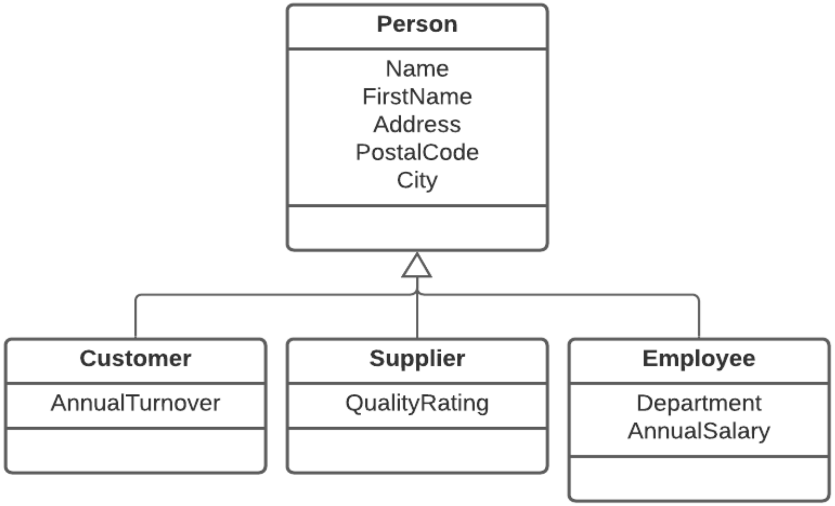
ERM versus OOD
OOD Representation

How are these entity types represented in an object oriented design in the development view?

ERM versus OOD
OOD Representation

How are these entity types represented in an object oriented design in the development view?

- While the use of primary and foreign keys is one of the means of expression of relational database design, the is-a relationship (subtyping or type inheritance) is a means of expression of object orientation.
- Such is-a relationships, expressed in relational databases with foreign keys, can be represented more directly between objects: A customer is a person.



ERM versus OOD
Approach Differences

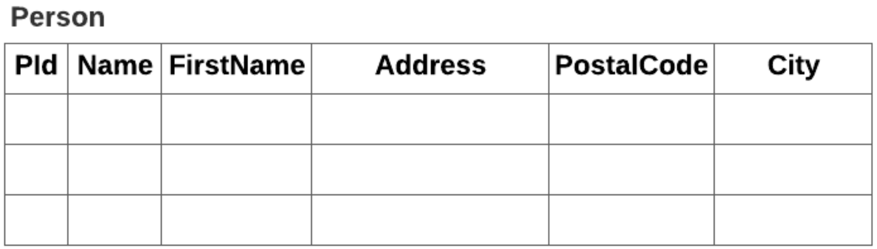
In the Logical View, differences between the two approaches become evident, especially with regard to when is-a relationships are established with concrete data.

The following scenario makes the differences visible:

1. customer Luis Müller, Bergweg 9, 8015 Exemplikon has an annual turnover of 284'295.
2. supplier Katja Maier, Hauptstrasse 525, 4711 Strasswil has a quality rating of 5.
3. employee Jonas Huber, Seitenstrasse 98, 2113 Waldbach works in the accounting department for an annual salary of 87’654.
4. Luis Müller, already known as a customer, is additionally employed in the shipping department with an annual salary of annual salary of 78,323.

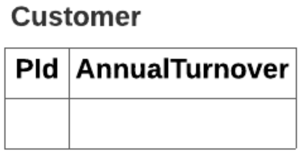
ERM versus OOD
Approach Differences

How are these entities represented in a relational database in the logical view?



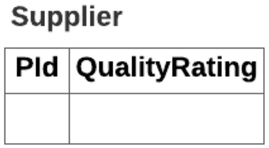
Person

PId	Name	FirstName	Address	PostalCode	City



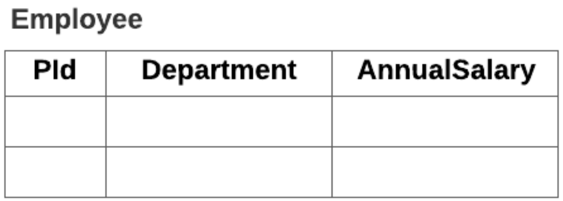
Customer

PId	AnnualTurnover



Supplier

PId	QualityRating

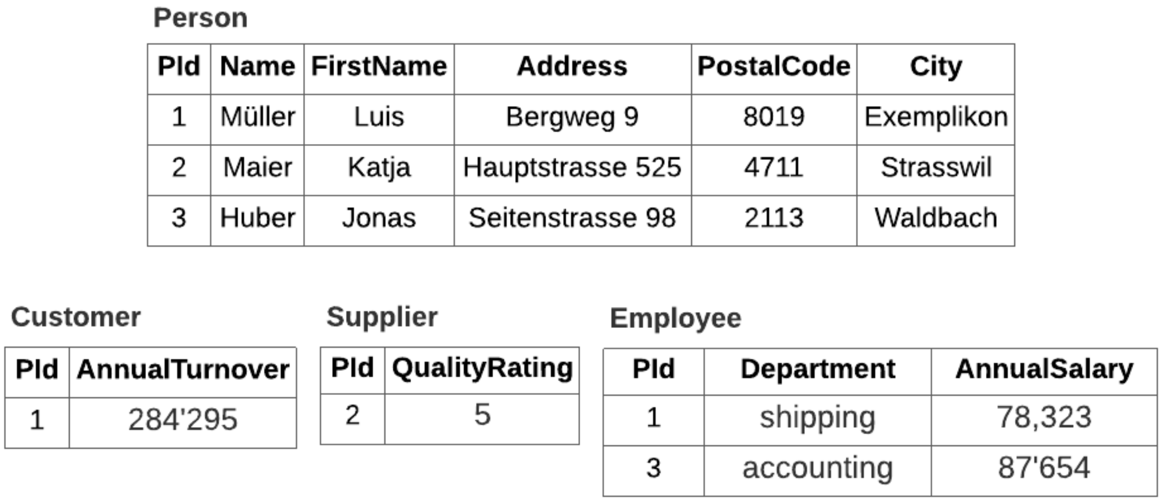


Employee

PId	Department	AnnualSalary

ERM versus OOD
Approach Differences

How are these entities represented in a relational database in the logical view?



Person

PId	Name	FirstName	Address	PostalCode	City
1	Müller	Luis	Bergweg 9	8019	Exemplikon
2	Maier	Katja	Hauptstrasse 525	4711	Strasswil
3	Huber	Jonas	Seitenstrasse 98	2113	Waldbach

Customer

PId	AnnualTurnover
1	284'295

Supplier

PId	QualityRating
2	5

Employee

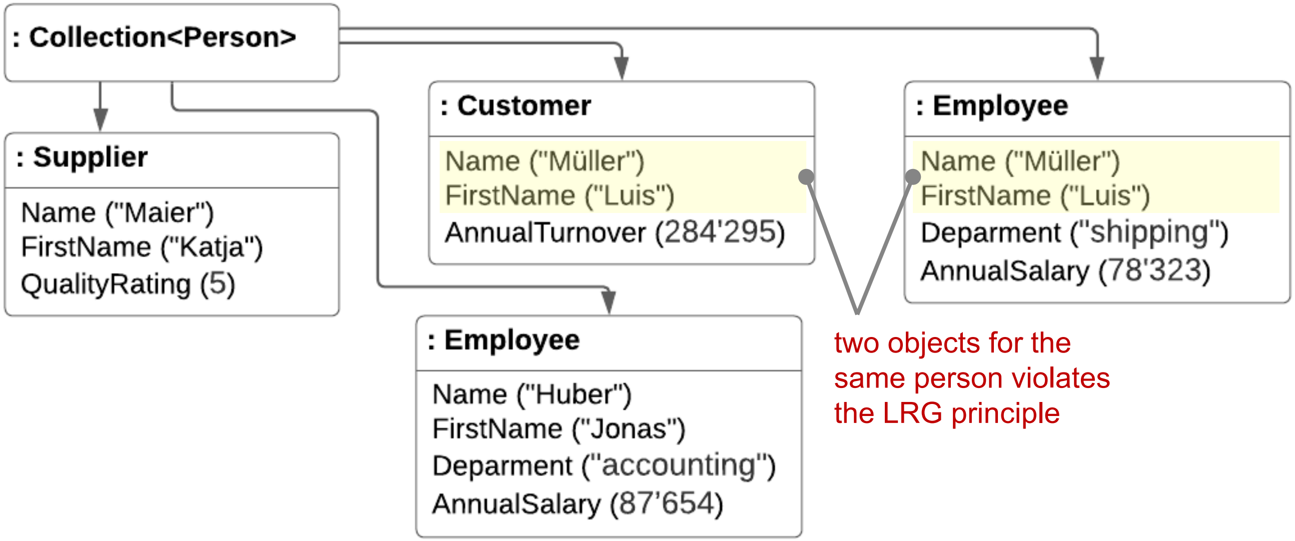
PId	Department	AnnualSalary
1	shipping	78,323
3	accounting	87'654

ERM versus OOD
Approach Differences

Now, how are the above entities represented in an object oriented design in the logical view?

ERM versus OOD
Approach Differences

Now, how are the above entities represented in an object oriented design in the logical view?



: Collection<Person>

: Supplier
Name ("Maier")
FirstName ("Katja")
QualityRating (5)

: Customer
Name ("Müller")
FirstName ("Luis")
AnnualTurnover (284'295)

: Employee
Name ("Müller")
FirstName ("Luis")
Depament ("shipping")
AnnualSalary (78'323)

: Employee
Name ("Huber")
FirstName ("Jonas")
Deparment ("accounting")
AnnualSalary (87'654)

two objects for the same person violates the LRG principle

ERM versus OOD

Approach Differences

Compare the two designs (i.e., ERM and OOD). What are the differences regarding to the associations of people with roles?

ERM versus OOD
Approach Differences

Compare the two designs (i.e., ERM and OOD). What are the differences regarding to the associations of people with roles?

- The ERM allows one *person* to have multiple *roles* at the same time. For the OOD, the respective classes for the different role combinations would have to be developed first.
- While the ERM allows a *role* to be added to or removed from a *person* at any time, the OOD specifies the *role* for a *person* at instantiation time.
- The OOD guarantees that a *person* object represents at most one *role*.

ERM versus OOD
Approach Differences

What does an object-oriented design look like that offers similar flexibility in the logical view as the relational database design above?

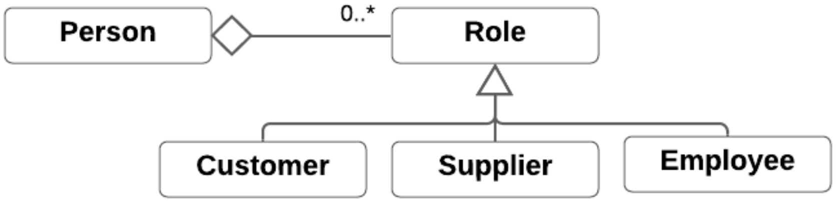
Development view

Logical view

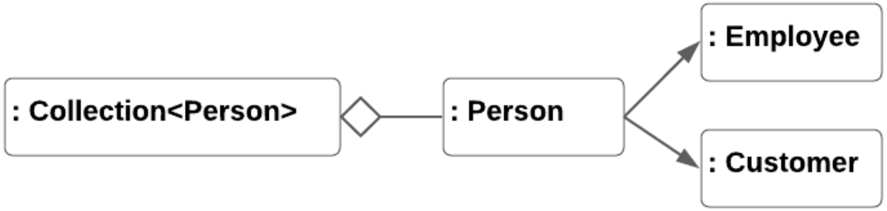
ERM versus OOD
Approach Differences

What does an object-oriented design look like that offers similar flexibility in the logical view as the relational database design above?

Development view



Logical view



Questions