Data Modeling, Entity-Relationship Diagram
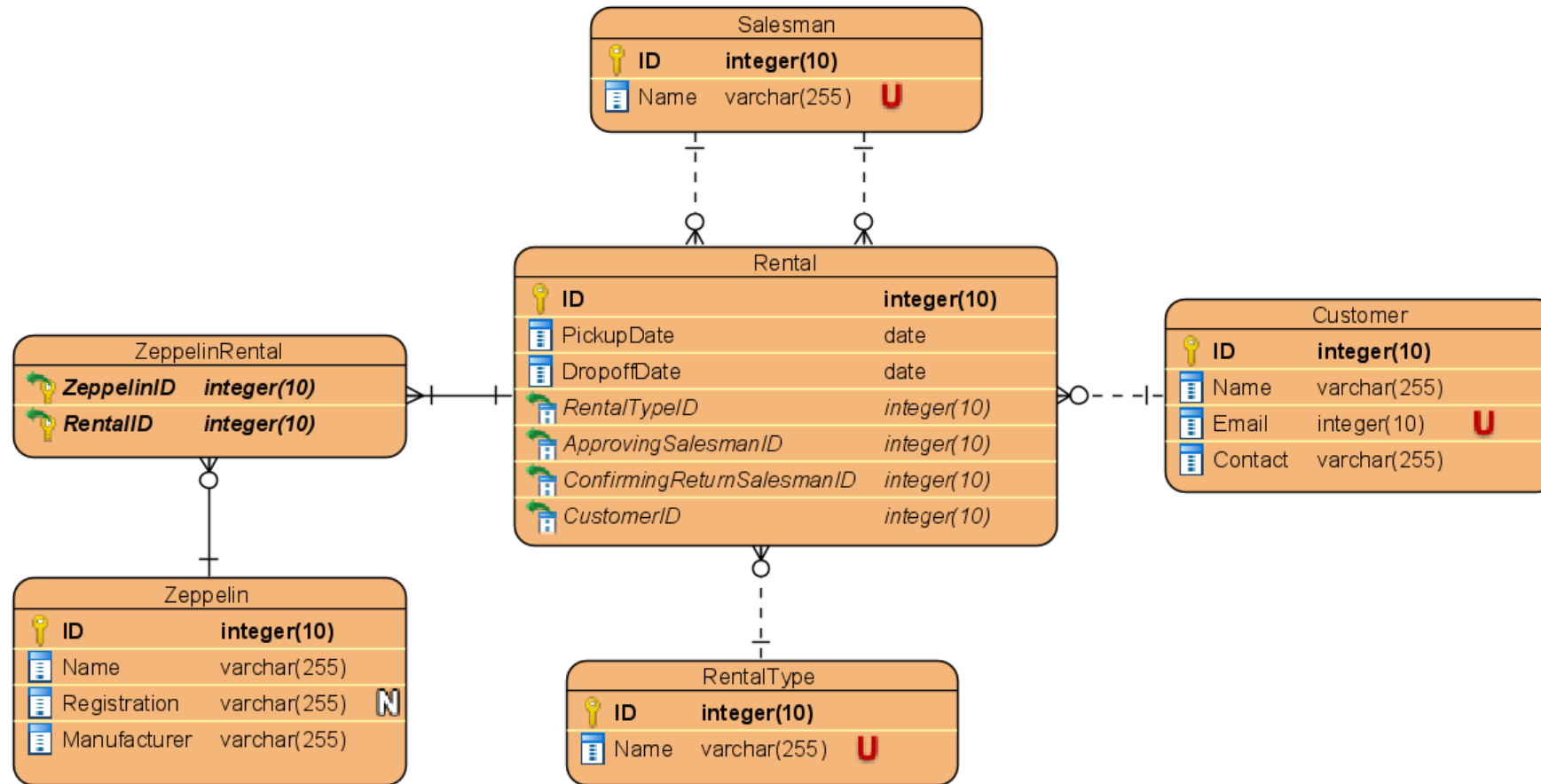
PB007 Software Engineering I

Lukáš Daubner

daubner@mail.muni.cz

Entity-Relationship Diagram

— Data model

— Not a part of UML

— Representing the logical structure of **relational database**

— Its main components are:
  — Entities
  — Relations
  — Attributes

MUNI
FI

Entity-Relationship Diagram

Two Worlds Collide

MUNI
FI

Two Worlds Collide

— World of Objects – Class Diagram

- Captures data and operations
- Classes are connected with different relationships with different semantics
- Objects have own dynamic lifecycle
- Manipulation through interaction

— World of Data – Entity-Relationship Diagram

- Captures just data
- Simple relationships
- Represents tables in relational database
- Manipulation through relational algebra

MUNI
FI

Object-Relational Mapping

– Conversion „between the worlds"
- – Persistent class ~ Entity type (table)
- – Object ~ Entity (table row)
- – Class attribute ~ Entity attribute (table column)
- – Association/Aggregation/Composition ~ Relation (connection via foreign keys)
- – Inheritance ~ … (manual work needed, see following slides)

– Mapping is not always 1:1!
- – Single class can be mapped to multiple tables
- – And vice versa
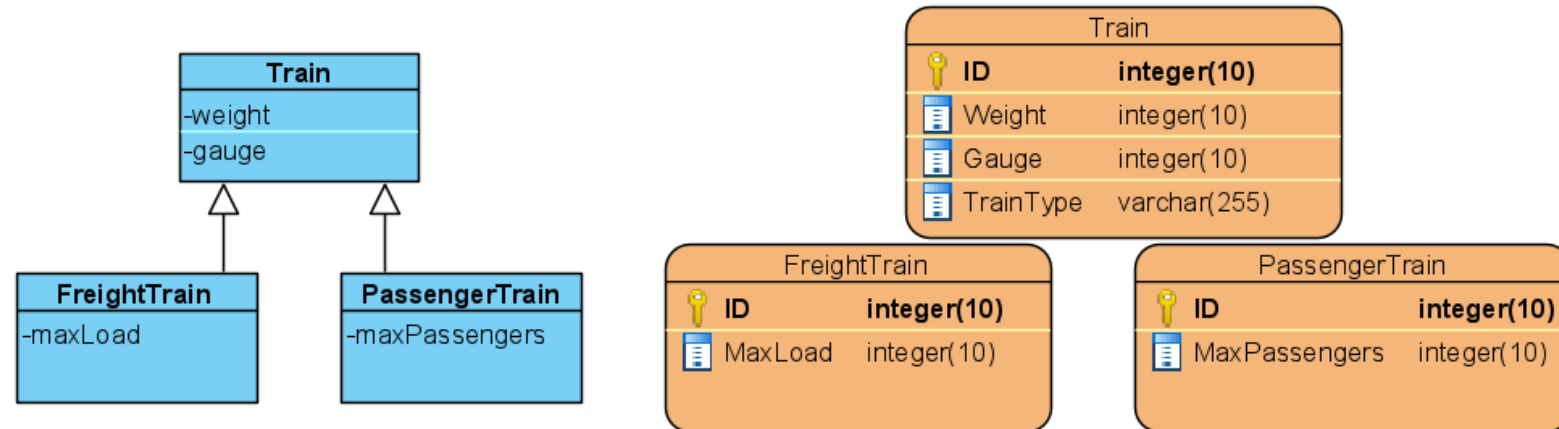- – Not all classes are **persistent** (objects stored in database)

MUNI
FI

Object-Relational Mapping

Example
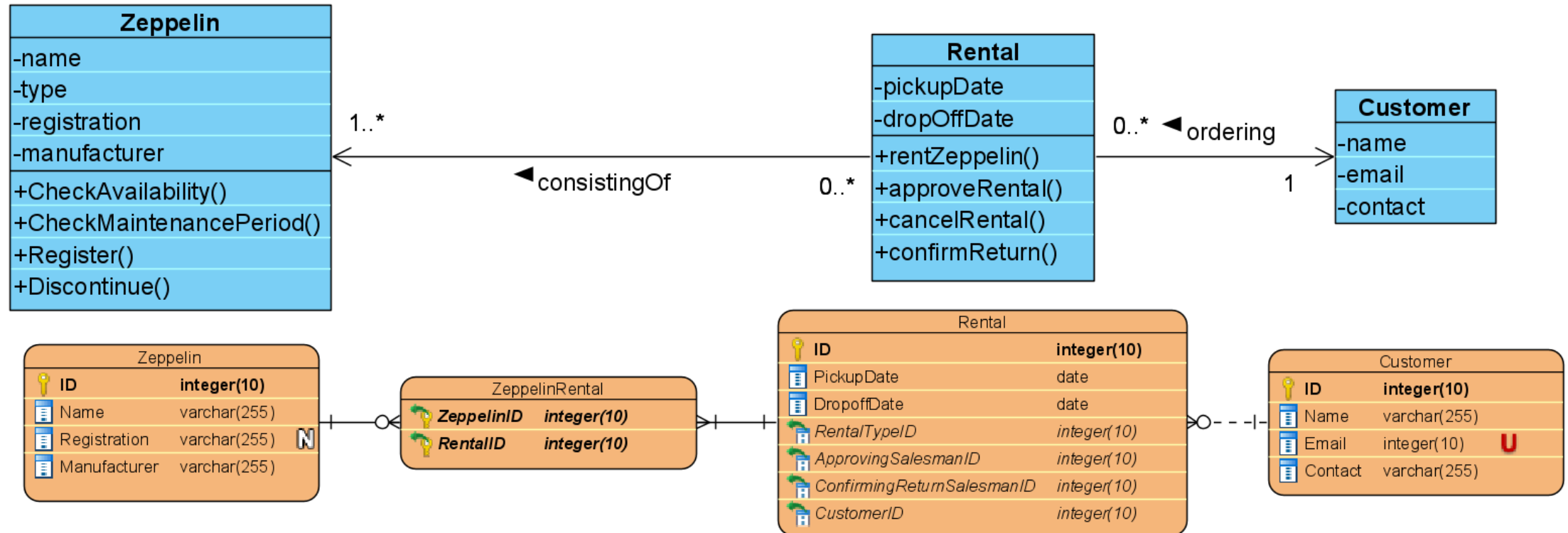
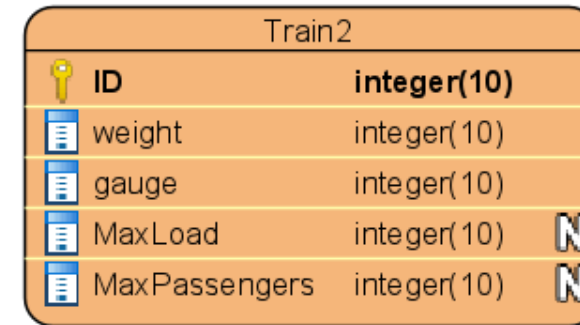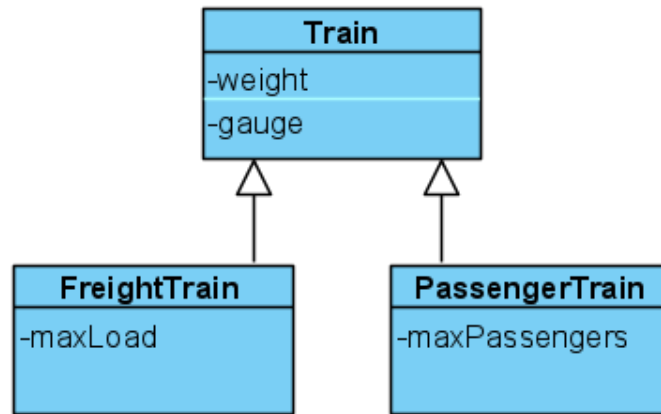# Object-Relational Mapping – Inheritance

1:1 Mapping

— Each class becomes a table

— An attribute differentiates the subclass type

— One object instance in multiple tables

   — More difficult data access

MUNI
FI

Object-Relational Mapping – Inheritance
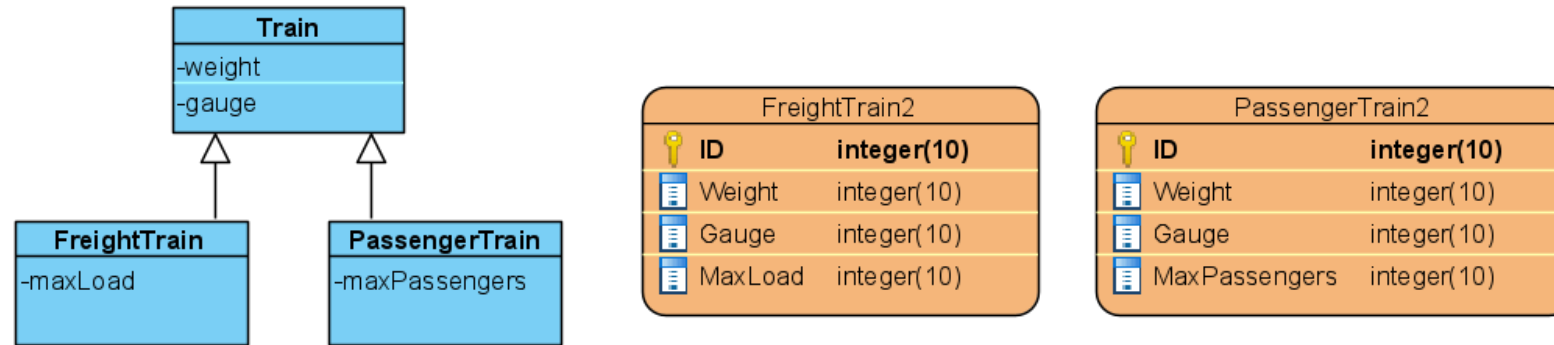
Merge to superclass

— All attributes in one table

— Some will have NULL value

  – Breaks the 4.NF

— Suitable for small number of subclasses and few attributes

MUNI
FI

Object-Relational Mapping – Inheritance

Propagation to subclasses

— Superclass attributes are copied to non-abstract subclass tables
— Suitable if:
  — Superclass has few attributes
  — Many subclasses
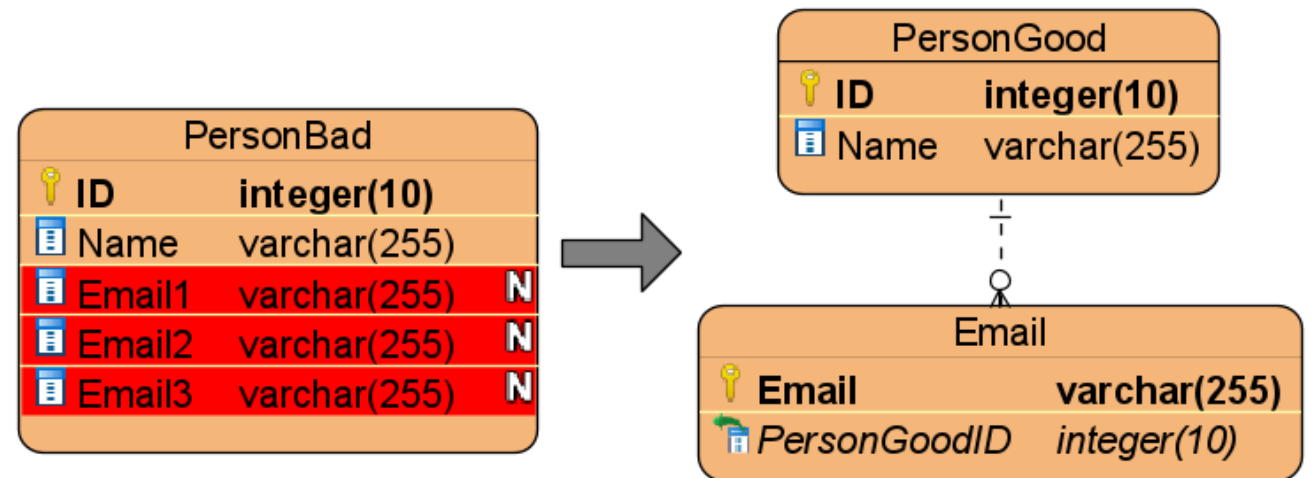 — Subclasses have many attributes

MUNI
FI

Normal Forms

— Technique for data organization and good database design

— Elimination of repetitive data

— Reduction of table complexity

— Problem prevention
  — E.g., update anomalies

MUNI
FI

Normal Forms

1. Normal Form

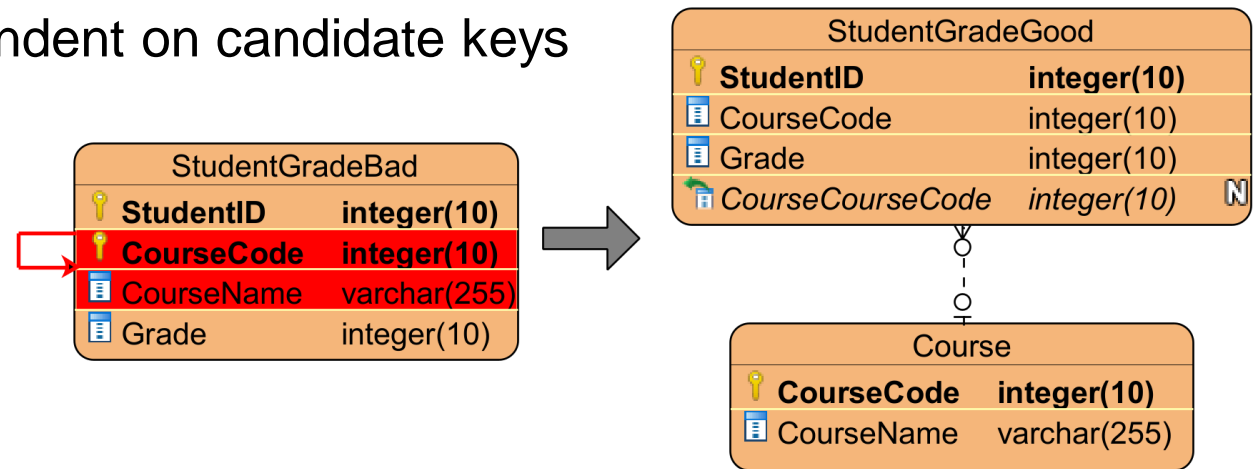— Satisfies 0. NF (yes, it actually exists)

— Each attribute is atomic

MUNI
FI

Normal Forms

2. Normal Form

— Satisfies 1. NF

— No partial dependency
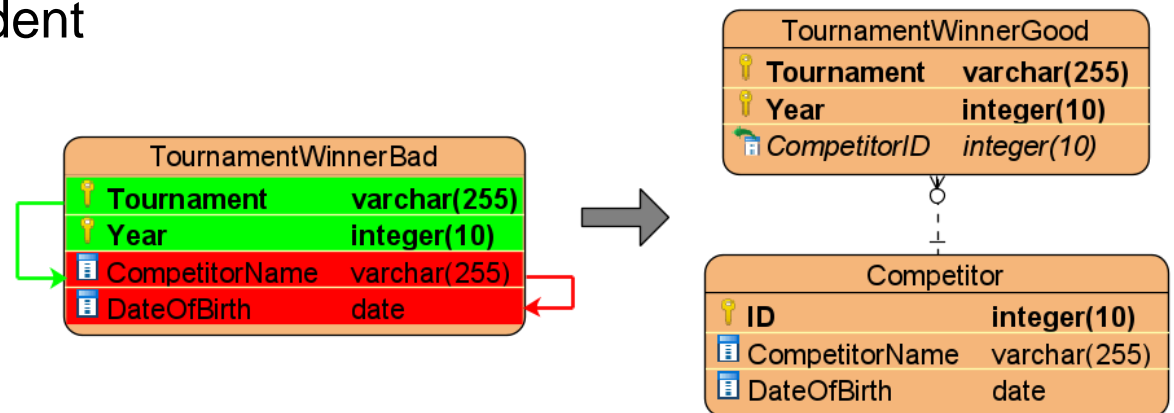 — Each non-key attribute are fully dependent on candidate keys

MUNI
FI

Normal Forms

— Satisfies 2. NF

— No transitive dependency

- Each non-key attribute is dependent on primary key (and candidate keys) only
- Non-key attributes are mutually independent

Task for this week

You gotta do what you gotta do

— Process the feedback

— Create ERD based on the class diagram
  — NB! Keep it consistent – you model the same system
  — Normalize to 3. NF

— Based on the EDR create separate example violating 1. and 2. NF

— Do your part in peer review
  — Link to roster is in study materials

MUNI
FI