Graz University of Technology

Object-Oriented Analysis & Design (OAD)

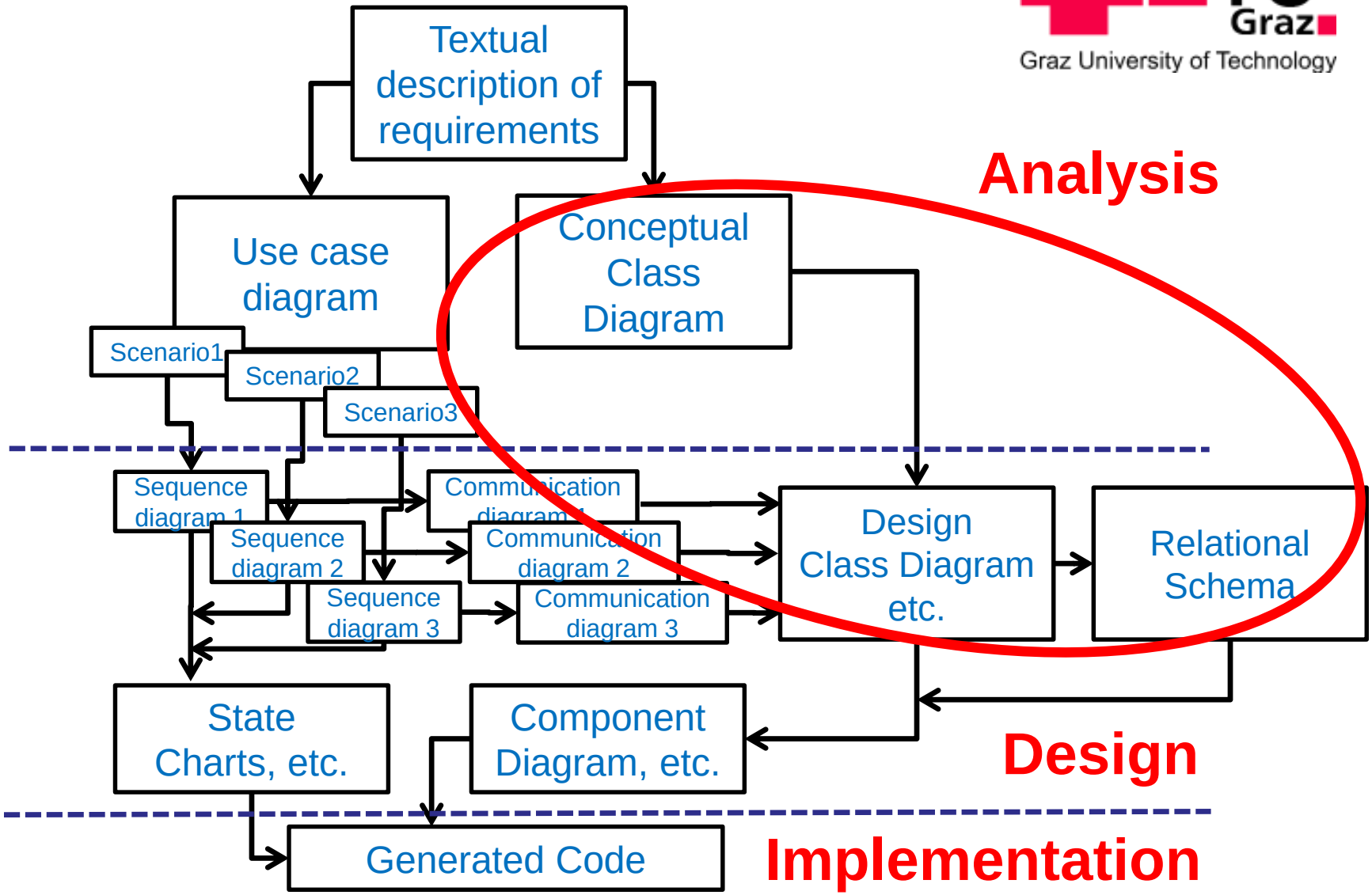
Object-Relational Mapping (ORM)

https://youtu.be/HorIWBNohEU

Alexander Felfernig
Institute for Software Technology
Inffeldgasse 16b/2

„Big Picture“

Graz University of Technology



Database Systems

Graz University of Technology

- Database (DB)
  - collection of data representing different aspects of the application domain
- Database management system (DBMS)
  - interface between user and DB
  - guarantees adequate and efficient data access, resistant to SW and HW failures, data available over a longer period
- Database system (DBS)
  - consists of a DBMS and one or more DBs

DBMS Languages

Graz University of Technology

- Data description (definition) language (DDL)
  - definition of database schema
- Data manipulation language (DML)
  - creating (INSERT), adapting (UPDATE), and deleting database contents (DELETE)
- Query language (QL)
  - querying DB contents (SELECT) without changing the content itself

Relational Database

Graz University of Technology

- Table (relation)
  - subset of tuples of the cross product of attribute domains together with column headings
  - e.g., persons ⟶

ID	firstname	lastname	date of birth
1	Albert	Einstein	14.03.1879
2	Nils	Bohr	07.10.1885
3	Werner	Heisenberg	05.12.1901
4	Max	Planck	23.04.1858
5	Christian	Doppler	23.11.1803
6	Ludwig	Boltzmann	20.02.1944
7	Erwin	Schrödinger	12.08.1887

- Attribute
  - ID, firstname, …
- Domain
  - set of values of the same type
- Tuple
  - element of the cross product of attribute domains
  - e.g., (1,'Albert','Einstein',14.03.1879)

Relational Database

Graz University of Technology

- Superkey of a table (relation) R
  - set of attributes of R which allow the unique identification of each tuple in R
  - for example: {orderID,orderDate} in table “order”
- Key of a table
  - superkey with “minimal” set of attributes
  - often artificial key, for example: {orderID} or simply {ID}
- Candidate key
  - one key of a relation R, where R can have different keys, for example, {custID}, {SSN}

Relational Database

Graz University of Technology

- Key attribute
  - attribute that belongs to a (candidate) key
  - for example: bookID in {authorID, bookID}
- Primary key
  - key used for identifying tuples in the table
  - for example: candidate keys {authorID} and {firstname, lastname, date of birth} → {authorID} selected as primary key
- Foreign key
  - set of attributes that occurs as primary key in another table (e.g., order.custID → customer.custID)

Relational Database

Graz University of Technology

- Referential integrity
  - each value (set of values) that occurs in table R as a foreign key instance referencing table R' must also exist as a primary key instance in R'
  - for example: order.custID = 5 → exists: customer.custID = 5
- Entity integrity
  - each tuple of R must have a complete primary key instance, i.e., there must be a value for each attribute of the primary key

Example Tables (1)

Graz University of Technology

- **customer**(custID, firstname, lastname, date of birth, address)
- **order**(orderID, date, *custID*)
- **orderpos**(posID, *orderID*, *productID*)
- **product**(productID, name, price)

- Used notation: **tablename**, primary key attribute, *foreign key attribute*

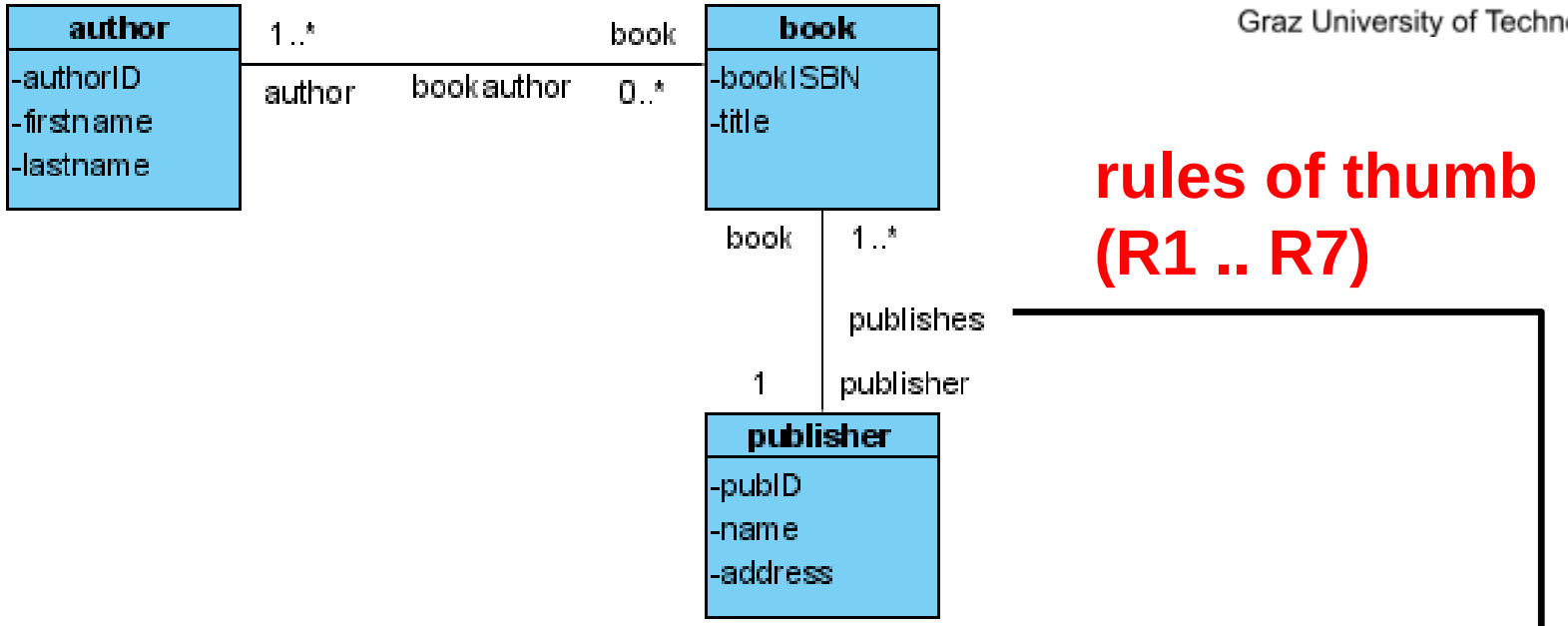
Example Tables (2)

Graz University of Technology

- **author**(authorID, firstname, lastname)
- **book**(bookISBN, title, *pubID*)
- **bookauthor**(*bookISBN*, *authorID*)
- **publisher**(pubID, name, address)

Database Mapping (ORM)

Graz University of Technology



rules of thumb (R1 .. R7)

- **author**(<u>authorID</u>, firstname, lastname)
- **book**(<u>bookISBN</u>, title, *publID*)
- **bookauthor**(<u>*bookISBN*</u>, <u>*authorID*</u>)
- **publisher**(<u>publID</u>, name, address)

R1 (Classes)

Graz University of Technology

- Rules R1..R7: translation of UML class diagrams to relational schemas
- For each class C (not association classes AC!) of the class diagram, a table C is created
  - table name = class name
  - all attributes of class C with cardinality max. 1 (~"single-valued") become attributes of table C
- Note: classes that are related to other classes by generalization might be treated differently (R7)
- Rules applied until no more Ri applicable

R1 (Example)

Graz University of Technology

- **author**(authorID, firstname, lastname)
- **book**(bookISBN, title, …)

R2 (Primary Key)

Graz University of Technology

- Identify the candidate keys in each table formed and specify one of them as primary (underline)

- Recommendation: if all candidate keys consist of more than one attribute, introduce an additional “artificial” attribute ID and specify it as primary key

R2 (Example)

- **author**(<u>authorID</u>, firstname, lastname)

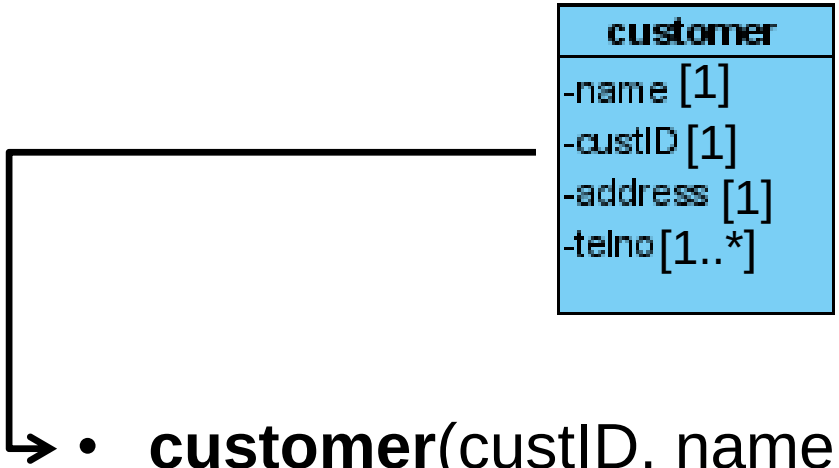
R3 (Multivalued Attribute)

Graz University of Technology

- For each attribute A of a class C (e.g. “customer”) with max. cardinality greater than 1 (e.g. “telno”) create a separate table:
  - table name = C_A (e.g. “customer_telno”)
  - primary key of C to be indicated as foreign key in C_A (e.g. *custID*)
  - attribute A (e.g. “telno”)
  - foreign key and A forms a minimal primary key of C_A (e.g. “*custID*, telno”)

R3 (Example)

Graz University of Technology



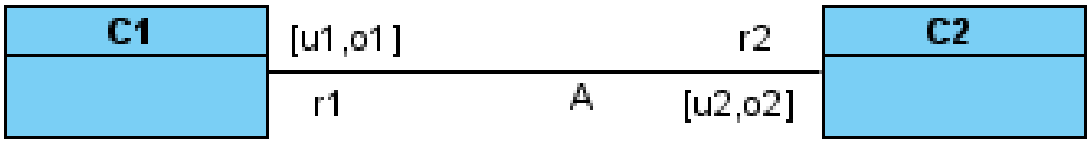
- **customer**(<u>custID</u>, name, address)
- **customer_telno**(*<u>custID</u>*, <u>telno</u>)

Association Connectivity

Graz University of Technology

- Given an association A between the classes C1 and C2, with roles r1, r2 and the multiplicities [u1,o1] and [u2,o2]
- Then A is called
  - **1:1 association** if o1 = 1 and o2 = 1
  - **1:n association** if o1 = 1 and o2 ≠ 1
  - **n:m association** if o1 ≠ 1 and o2 ≠ 1

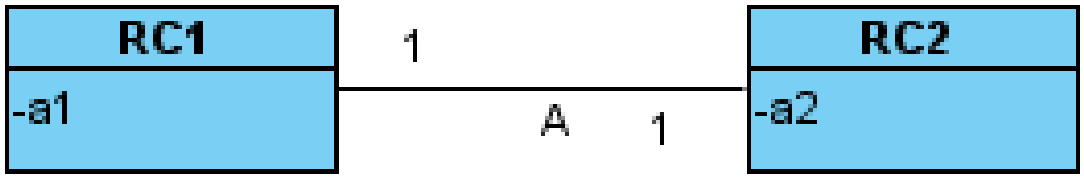
} **connectivities**



R4 (1:1 association)

Graz University of Technology

- An 1:1 association A between the classes RC1 and RC2 is mapped by extending the resulting tables RC1 and RC2 by the attributes a1 and a2 (foreign keys)



- **RC1**(<u>a1</u>)
- **RC2**(<u>a2</u>)

→

- **RC1**(<u>a1</u>, *a2*)
- **RC2**(<u>a2</u>, *a1*)

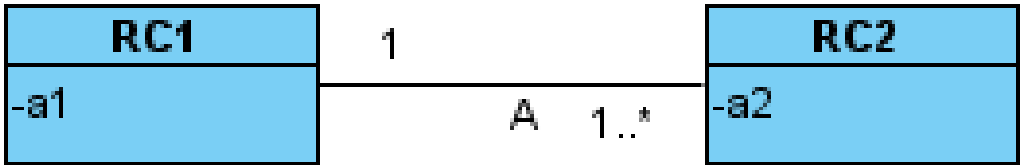
R4 (Example)

- **department**(<u>deplD</u>, name, *emplD*)
- **employee**(<u>emplD</u>, name, *deplD*)

R5 (1:n association)

Graz University of Technology

- An 1:n association A is mapped by extending the resulting table RC2 by the attribute a1 as foreign key

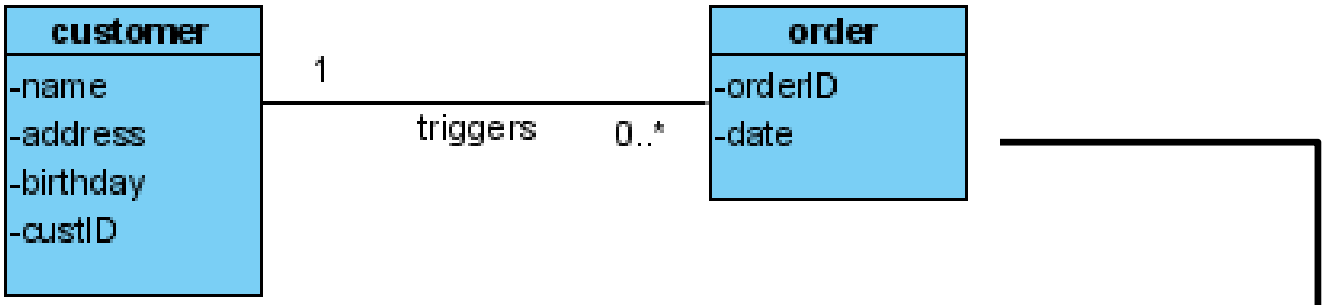


- **rc1**(<u>a1</u>)
- **rc2**(<u>a2</u>)

→

- **rc1**(<u>a1</u>)
- **rc2**(<u>a2</u>, *a1*)

R5 (Example)

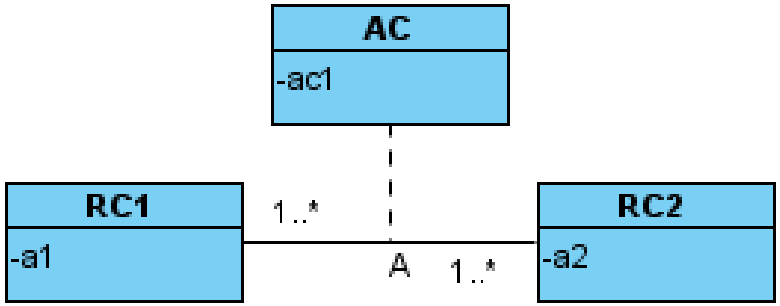


- **customer**(<u>custID</u>, name, address, birthday)
- **order**(<u>orderID</u>, date, *custID*)

R6 (m:n association)

Graz University of Technology

- An **m:n association A** with the attributes ac1, ac2, …, acn **between the classes RC1 and RC2** is mapped by creating the table A that includes aci and the primary key attributes of RC1 and RC2 as foreign keys



- **RC1**(<u>a1</u>)
- **RC2**(<u>a2</u>)

→

- **RC1**(<u>a1</u>)
- **RC2**(<u>a2</u>)
- **A**(<u>*a1*</u>, <u>*a2*</u>, ac1)

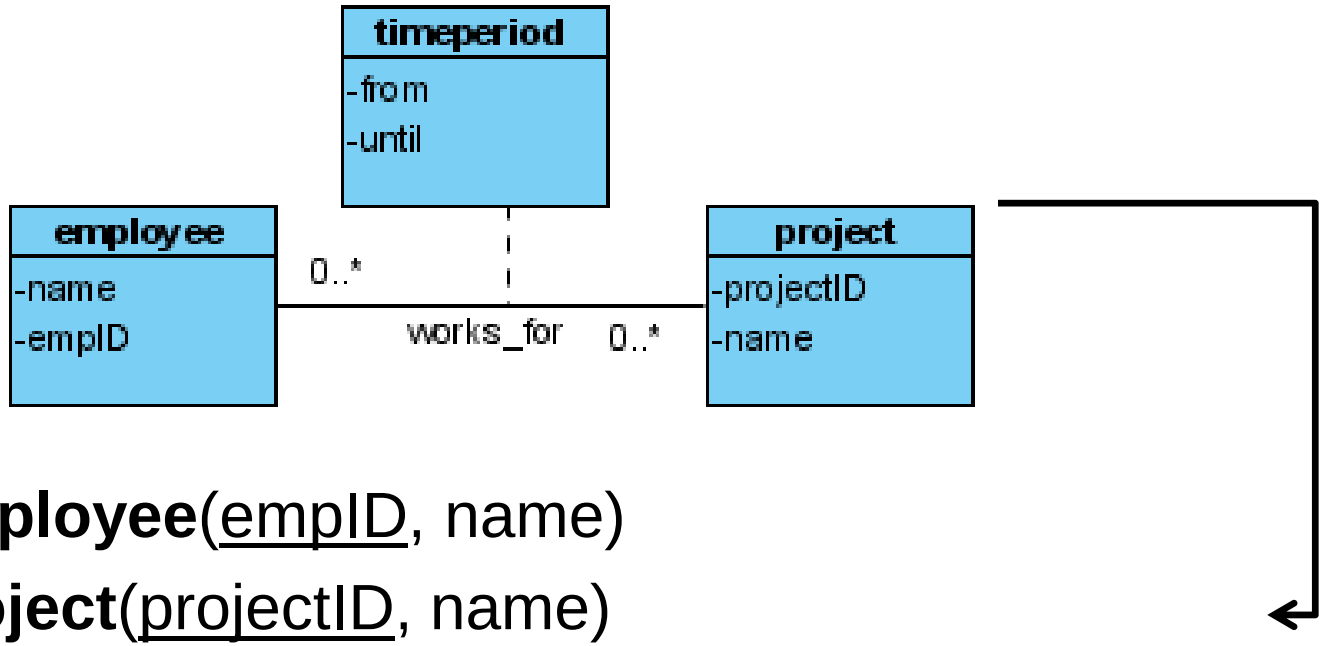
R6 (Example 1)

Graz University of Technology

- **author**(<u>authorID</u>, firstname, lastname)
- **book**(<u>bookISBN</u>, title, publisher)
- **bookauthor**(*<u>authorID</u>*, *<u>bookISBN</u>*)

R6 (Example 2)

Graz University of Technology



- **employee**(empID, name)
- **project**(projectID, name)
- **works_for**(*empID*, *projectID*, **from, until**)**?**
- **works_for**(*empID*, *projectID*, **from, until**)**?**
- **works_for**(ID, *empID*, *projectID*, from, until)

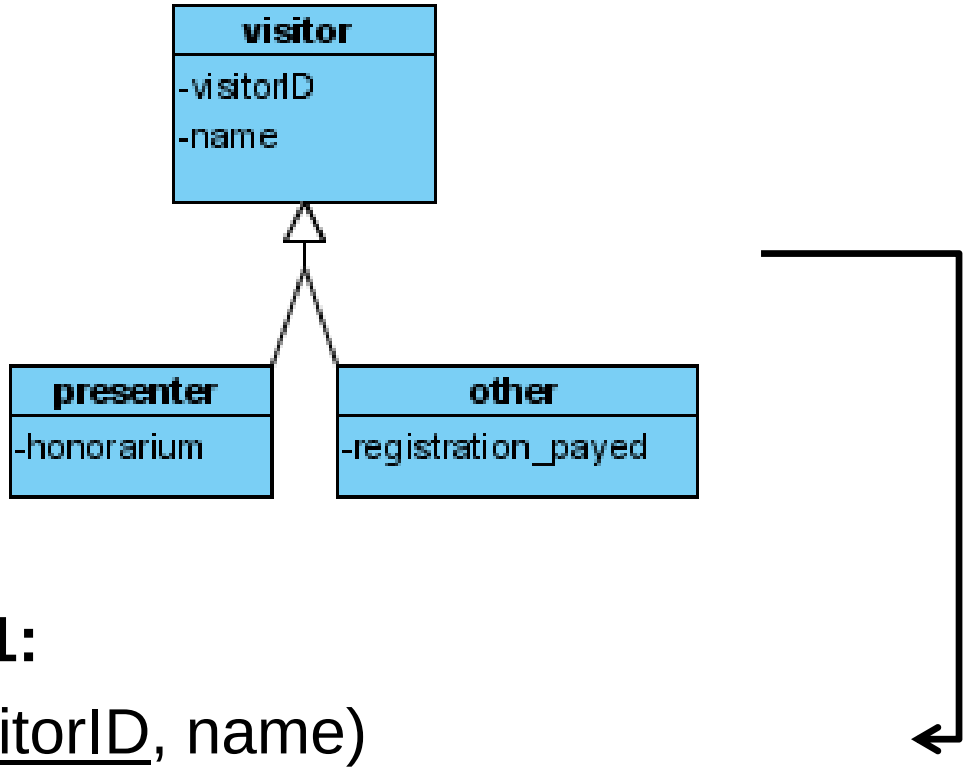
Rule 7 (Generalization)

Graz University of Technology

- A generalization relationship between RCs and RCg can be mapped as follows:
- **Alternative 1 (ALT1)**:
  - One table per class: extend RCs with the primary key of RCg (as foreign key)
- **Alternative 2 (ALT2)**:
  - One table per concrete class: extend RCs with all attributes from RCg and delete RCg
- **Alternative 3 (ALT3)**:
  - One table per generalization hierarchy: extend RCg by the attributes of RCs and delete RCs

R7 (Example – ALT1)

Graz University of Technology

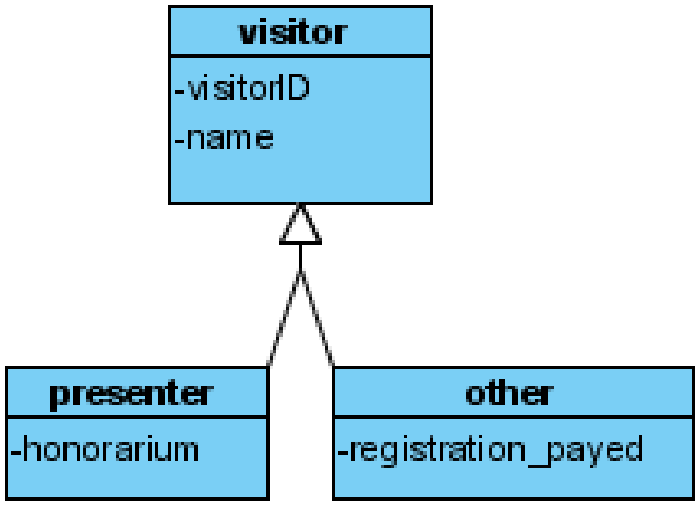


Alternative 1:

- **visitor**(visitorID, name)
- **presenter**(*visitorID*, honorarium)
- **other**(*visitorID*, registration_payed)
- JOINS (-) but generalization hierarchy (+)

R7 (Example – ALT2)

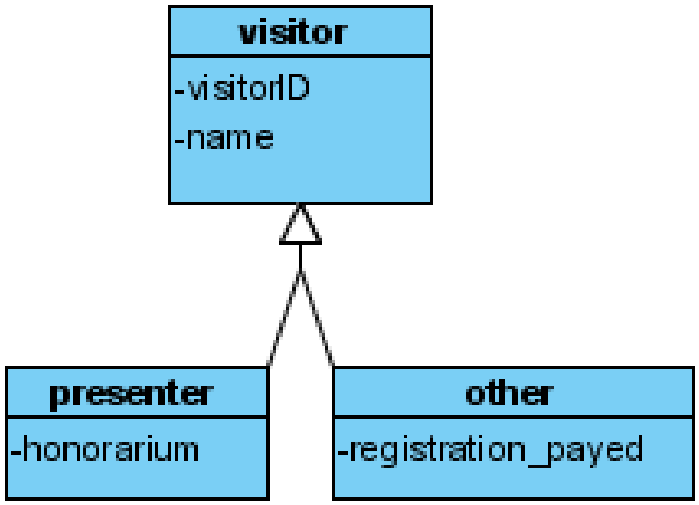
Graz University of Technology



Alternative 2:

- **presenter**(visitorID, name, honorarium)
- **other**(visitorID, name, registration_payed)
- No JOINS (+) but schema redundancy (-)

R7 (Example – ALT3)

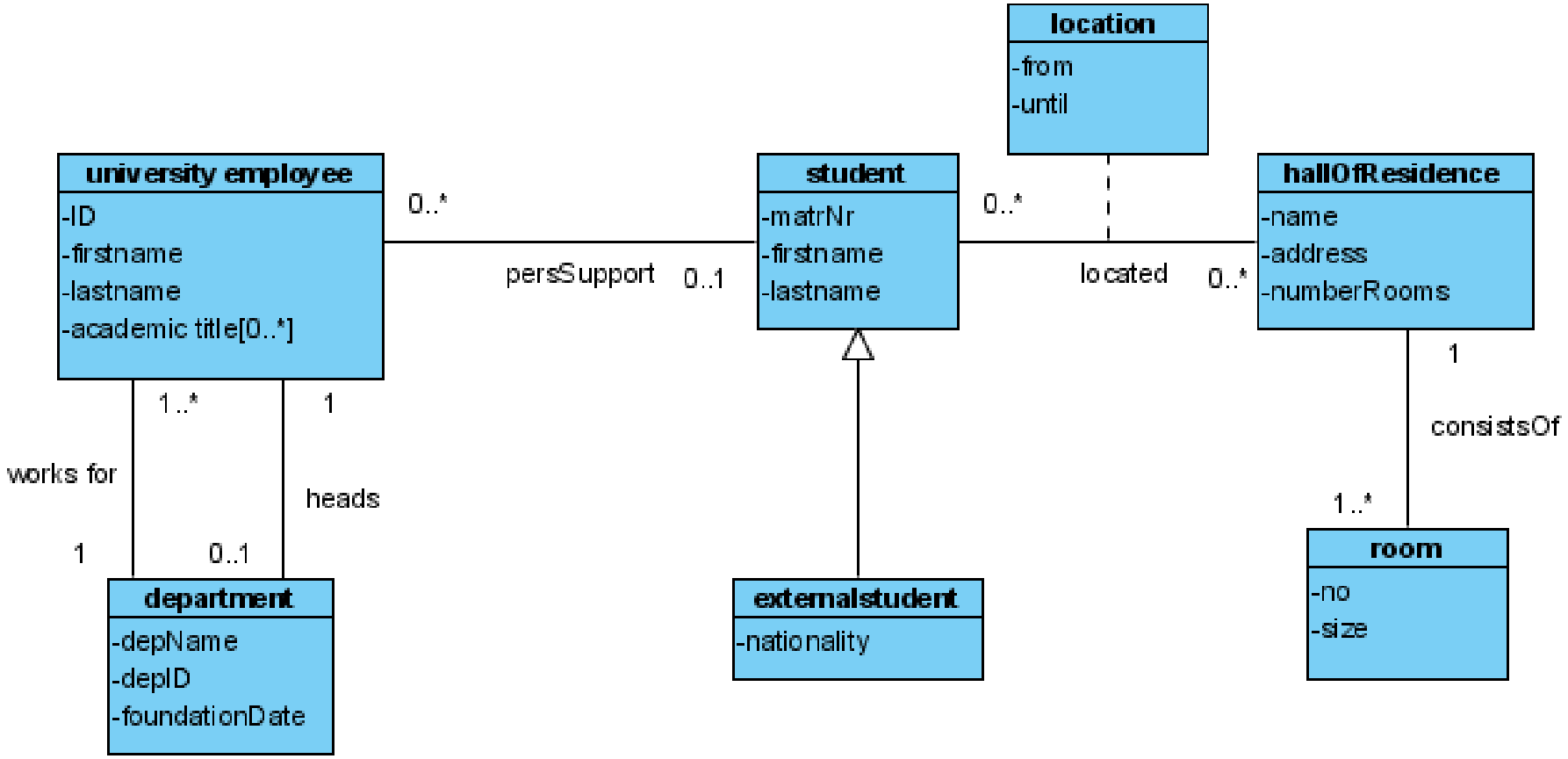


Alternative 3:

- **visitor**(visitorID, name, honorarium, registration_payed)
- No JOINS (+) but more memory needed (-)

Integrated Example (Class Diagram)

Graz University of Technology



Integrated Example (Database Structure)

Graz University of Technology

- **universityemployee**(ID, firstname, lastname, *dep_w, dep_h, matrNr*)
- **universityemployee_academictitle**(academictitle, *ID*)
- **department**(depID, depName, foundationDate, *ID*)
- **student**(matrNr, firstname, lastname)
- **externalstudent**(*matrNr*, nationality)
- **hallOfResidence**(name, address, numberRooms)
- **room**(no, size, *name*)
- **located**(LID, *matrNr*, *name*, from, until)

Graz University of Technology

Thanks!

ase.ist.tugraz.at
www.felfernig.eu