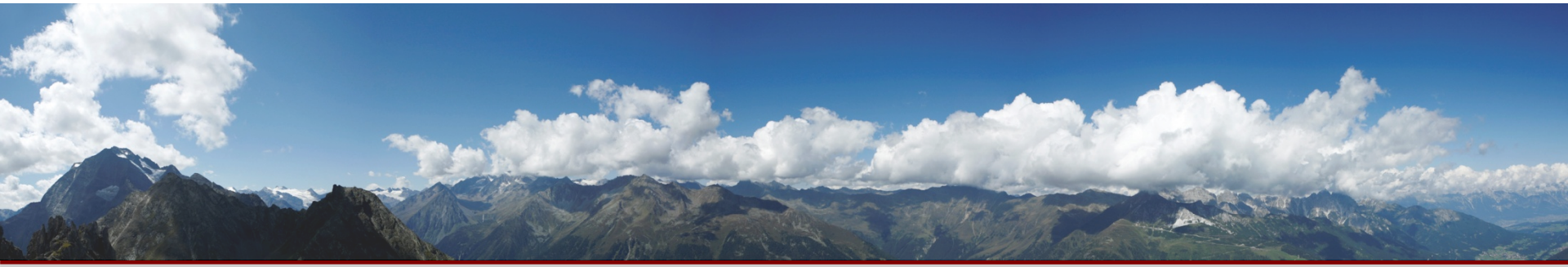
Introduction to modeling
WS 2015/16

Relational modelling

Slides for this part are based on
Chapters 11 from Halpin, T. & Morgan, T. 2008, Information Modeling and Relational Databases, Second Edition (ISBN: 978-0-12-373568-3), published by Morgan Kaufmann Publishers.



Where are we?

STI · INNSBRUCK

#	Title	Date
1	Introduction	07.10.2015
2	General concepts	21.10.2015
3	ORM modeling	04.11.2015
4	**Relational modeling**	**18.11.2015**
5	ER modeling	02.12.2015
6	OO modeling	16.12.2015
7	Services and process modeling	13.01.2016
10	Exam	27.01.2016

Relational model

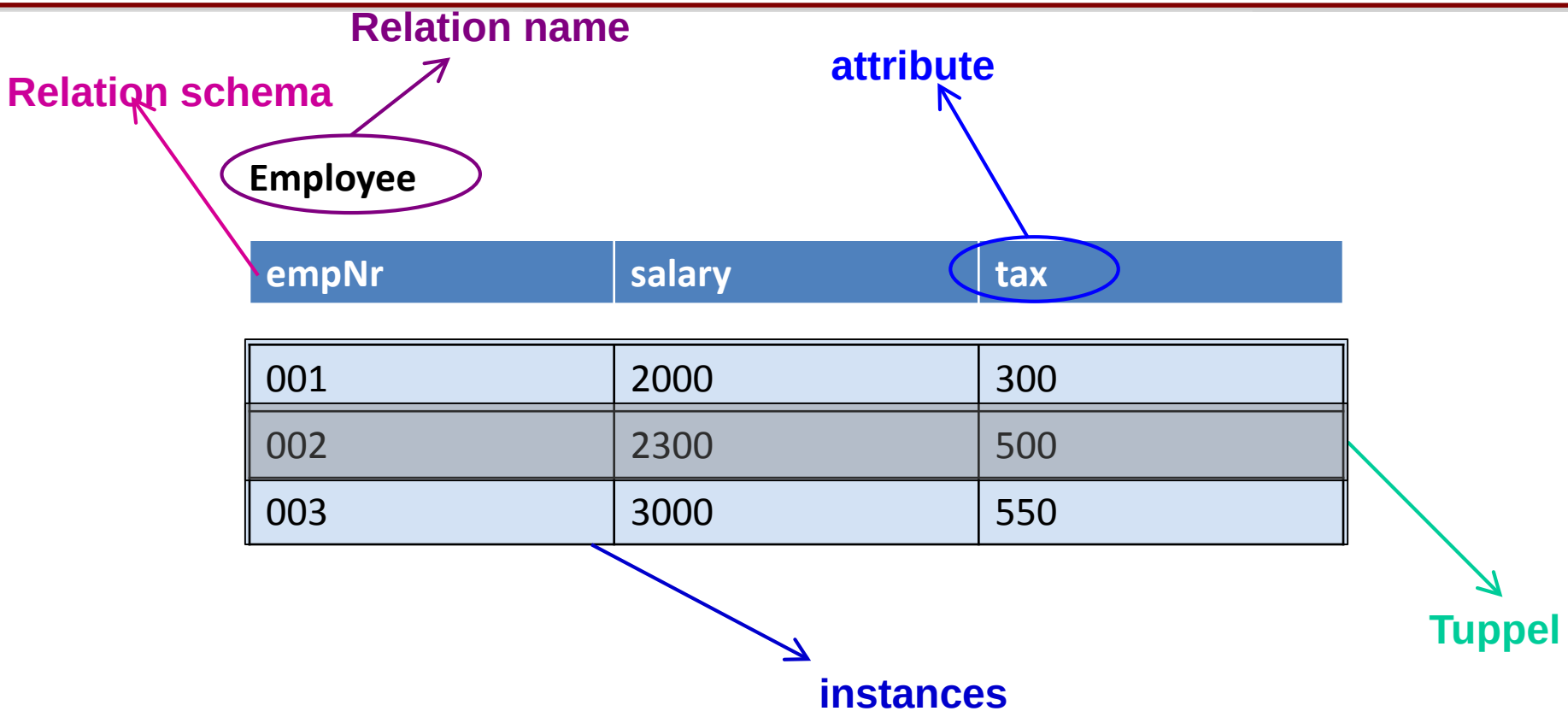
STI · INNSBRUCK

- Data Model – Set of concepts to describe the structure of a database
- Relational Model – The database can be regarded as a collection of tables
- Key concepts:
  - Relation: A mathematical concept that can be interpreted as a table of values
  - Relational Database: A collection of relations
    null indicates that no value is stored in that position

Terminology

- Domain: An set of atomic values.
  - Atomic means that items can not themselves be sets
- Attribute: A name of a role played by a domain (column name)
  - If A is an attribute, we write dom(A) = D to express that A is a role played by the domain D
- Relational schema $R(A_1,A_2, \ldots ,A_n)$: A named set of attributes $R = \{A_1,A_2, \ldots ,A_n\}$ where R is the relation name
  - n is the arity of the relationship
- Instance of a relational schema $R(A_1,A_2, \ldots ,A_n)$: A set $\{t_1,t_2, \ldots ,t_m\}$ ("rows") where each t_k is a n-tuple of values from the domain of $A_1,A_2, \ldots ,A_n$
- Relation: A relational schema with an associated instance

Example



Horizontal layout notation: the table name precedes a parenthesized list of column names, separated by commas:

Employee (empNr, salary, tax)

Remarks

STI · INNSBRUCK

- Tuples' order in relations is arbitrarily
- The order of values within a tuple is not arbitrary
- In one relations there cannot be two identical tuples
- A domain can be finite or infinite
- Two attributes in a relational schema can have the same domain, but not the same name

Uniqueness constraints on relational columns

STI · INNSBRUCK

- In horizontal layout, **uniqueness constraints** on relational columns are shown by underlining.
- Each unique column, or unique column combination, provides a ***candidate key*** for identifying rows in a table.
  - A key is a *minimal* set of uniquely constrained attributes, i.e. if an attribute is removed from a compound key, the remaining attributes are not spanned by a uniqueness constraint.
- If there is only one candidate key, this is automatically the ***primary key***, e.g.
  - *Employee* (empNr, salary, tax)

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

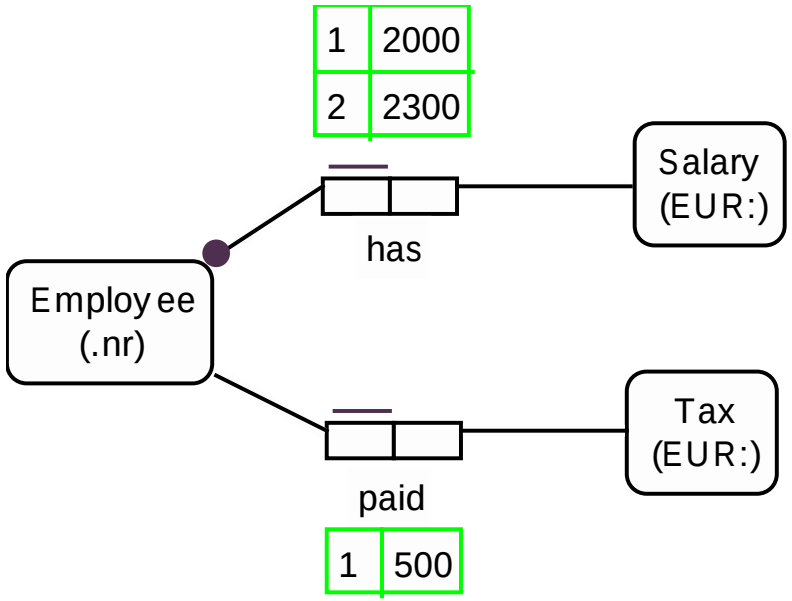
Uniqueness constraints on relational columns

STI · INNSBRUCK

- If more than one candidate key exists, one of these must be selected as the primary key. The others are then called *"alternate keys"* or *"secondary keys".* Primary keys are doubly underlined if alternate keys exist, e.g.
  - *Employee* (empNr, empName, deptCode, salary, tax)
- A column that does not allow null values is said to be **mandatory** (A column that does allow null values is said to be *optional.* Optional columns are enclosed in *square brackets, e.g.*
  - *Employee* (empNr, salary, [tax])

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Mapping ORM models



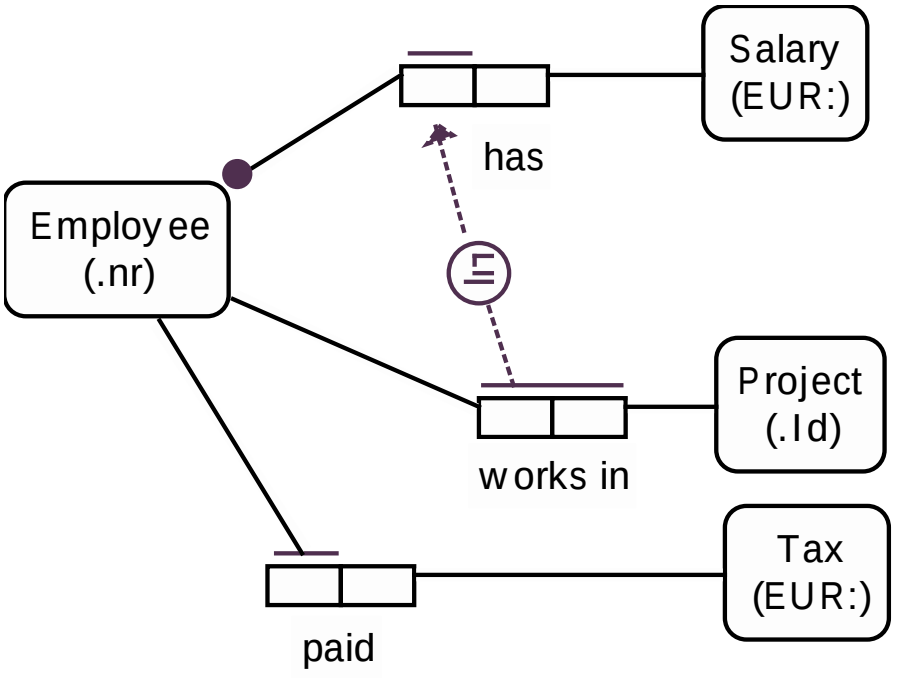
=> *Employee* (empNr, salary, [tax])

1	2000	500
2	2300	?

- Relational model often requires different facts about the same object to be stored in different tables
- Mandatory role constraints are captured by making their columns mandatory in their table, and running a subset constraint from other tables (if any) that contain facts about that object type

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Mapping ORM models



=>

Employee (<u>empNr</u>, salary, [tax])

WorksIn (<u>empNr, project</u>)

empNr in the WorksIn table is a **foreign key**

Integrity rules

- *Entity integrity rule:* primary keys contain no nulls (i.e., each column in a primary key is a mandatory column for its table).

- *Referential integrity rule*: each non-null value of a foreign key must match the value of some primary key

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

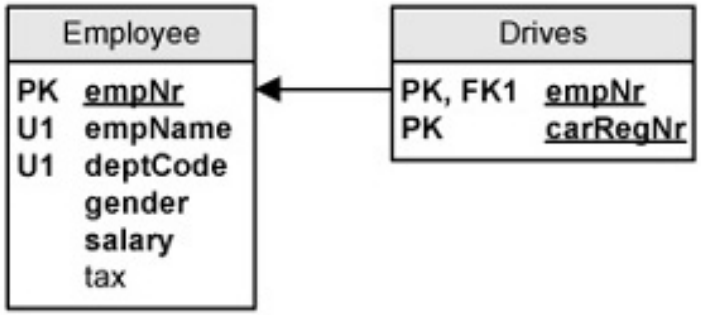
Relational schema representations

Relational schema in horizontal layout and some vertical layouts:

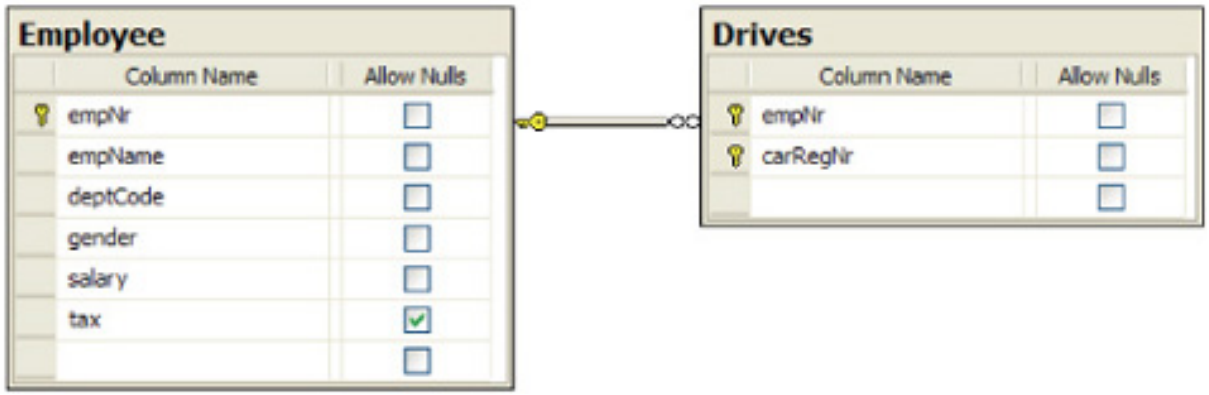
(a) *Employee* (empNr, empName, deptCode, gender, salary, [tax])
{M,F}

Drives (empNr, carRegNr)
≤ 3

(b)



(c)



Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Relational schema representations

STI · INNSBRUCK

Relational schema in SQL:

```
create domain EmpNr smallint;
create domain Money decimal(9,2);

create table Employee (
  empNr          EmpNr          notnull     primary key,
  empName     varchar(20) not null,
  deptCode     varchar(5)    not null,
  gender          char              not null    check(gender in ('M','F')),
  salary           Money         not null,
  tax               Money,
  unique ( deptCode, empName ) );

create table Drives (
  empNrEmpNr                  not null    references Employee,
  carRegNr     char(6)          not null,
  primary key ( empNr, carRegNr ),
  check (not exists
            (select empNr from Drives
             group by empNr
             having count(*) > 3));
```

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Relational Mapping Procedure Rmap

STI · INNSBRUCK

- Several different relational designs may be possible for a given conceptual schema
  - But the resulting relational design should be correct, efficient, and clear
- A high priority is placed on simplifying the enforcement of constraints at update time
  - Avoid redundancy, but this can lead to more tables in the design
  - For efficiency, number of tables should be kept down to an acceptable limit
- **Rmap** procedure guarantees a redundancy-free relational design and includes strategies to restrict the number of tables

Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Rmap – basic rules

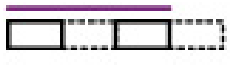
STI · INNSBRUCK

- Typically each row of a relational table stores one or more elementary facts
- Avoiding redundancy in tables: ensure that *each fact type maps to only one table in such a way that its instances appear only once*

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

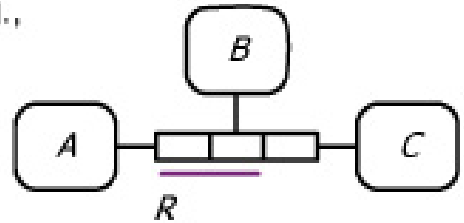
Rmap – basic rules

- Two basic rules:
  1. Each fact type with a compound, internal uniqueness constraint maps to a separate table by itself, keyed on the uniqueness constraint



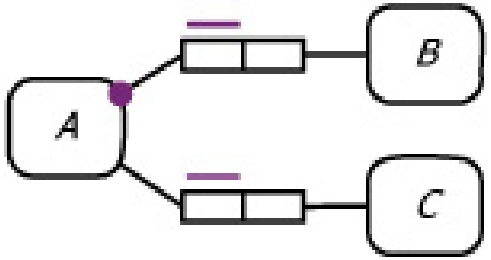
⇨ separate table

e.g.,



⇨ R (a, b, c)

  2. Fact types with functional roles (i.e. roles that have a simple uniqueness constraint) attached to the same object type are grouped into the same table, keyed on the object type's identifier

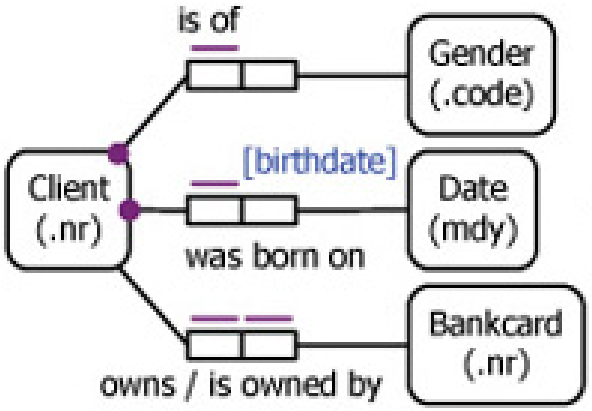


⇨ A (a, b, [c])

Material on this slide based on *Ch 11.2* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

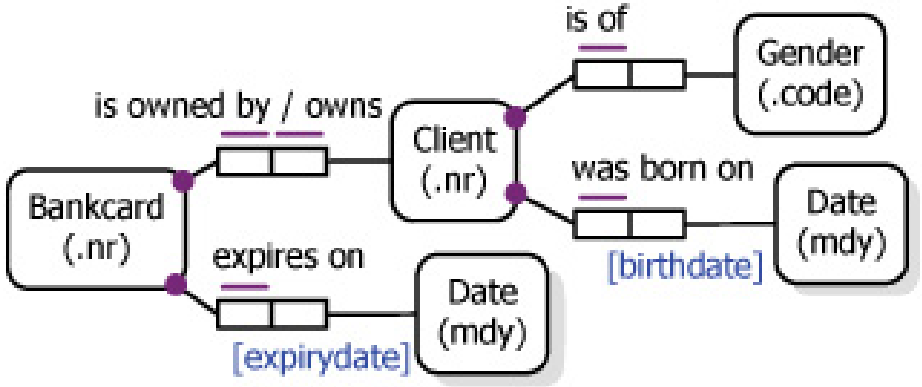
Rmap - Mapping 1:1 Associations

When no additional functional roles:



Client (clientNr, gender, birthdate, [bankcard])

When additional functional roles:



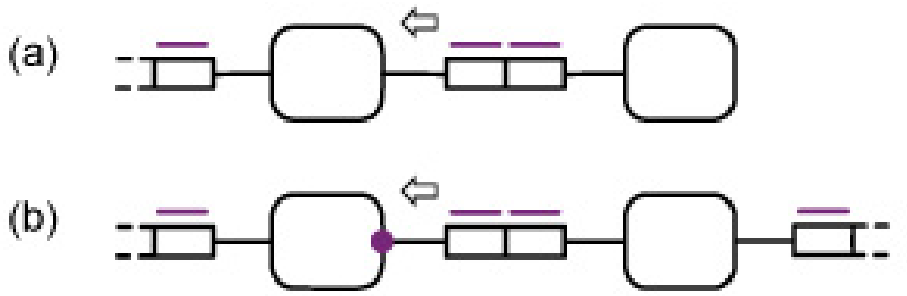
Client (clientNr, gender, birthdate)

Bankcard (cardNr, clientNr, expirydate)

Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Rmap - Mapping 1:1 Associations

Procedure:



Each *1:1 fact type* maps to only one table:

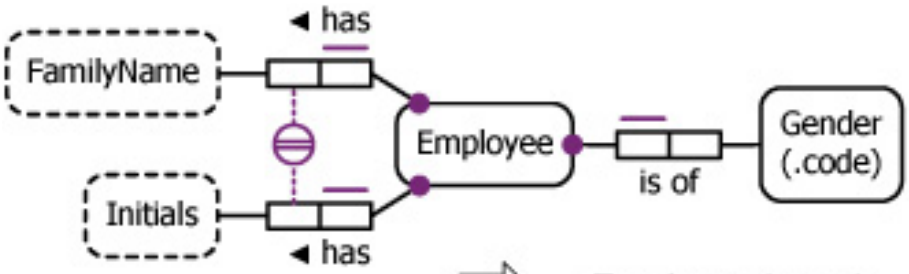
if (a) only one object type in the 1:1 predicate has another functional role
then group on its side
else if (b) both object types have other functional roles
and only one role in the 1:1 is explicitly mandatory
then group on its side
else if no object type has another functional role
then map the 1:1 to a separate table
else the grouping choice is up to you

Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Rmap – Mapping External Uniqueness Constraints

STI · INNSBRUCK

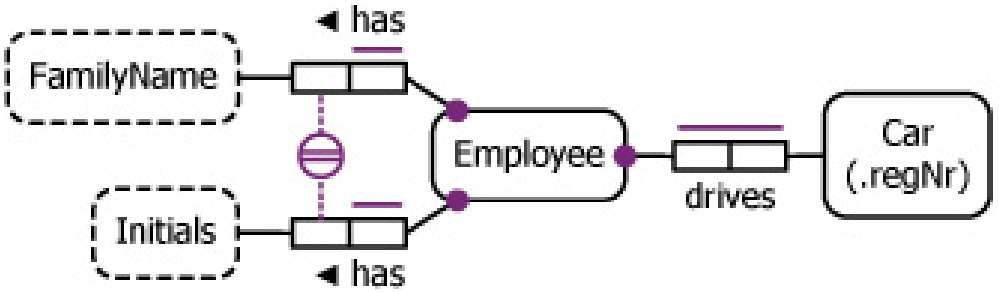
Composite primary identifier, and functional fact type:



⇨ *Employee* (e, g)

⇨ *Employee* (familyName, initials, gender)

Composite primary identifier and nonfunctional fact type:



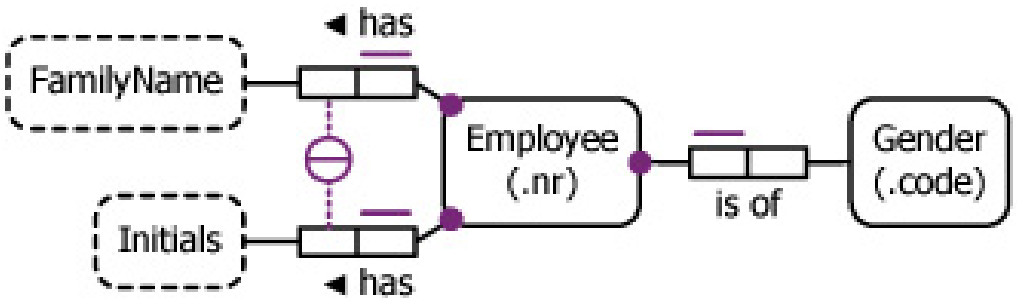
⇨ *Drives* (e, c) ⇨ *Drives* (familyName, initials, carRegNr)

Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Rmap – Mapping External Uniqueness Constraints

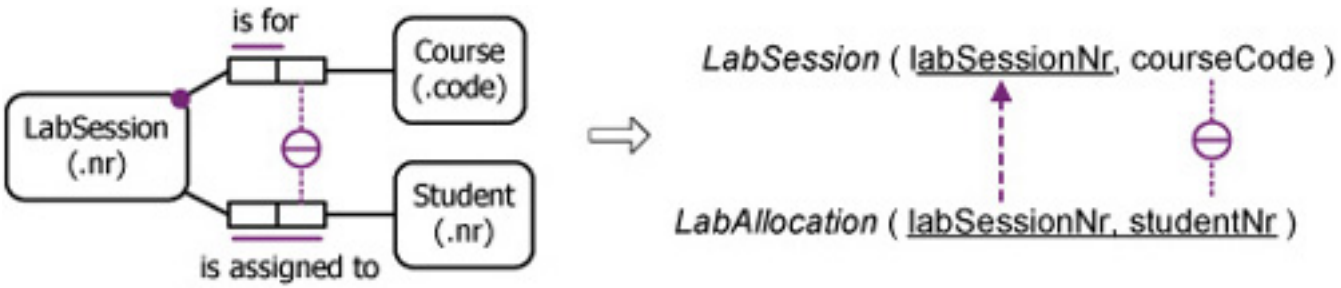
STI · INNSBRUCK

Composite secondary identifier and functional fact type:



⇨ *Employee* (empNr, familyName, initials, gender)

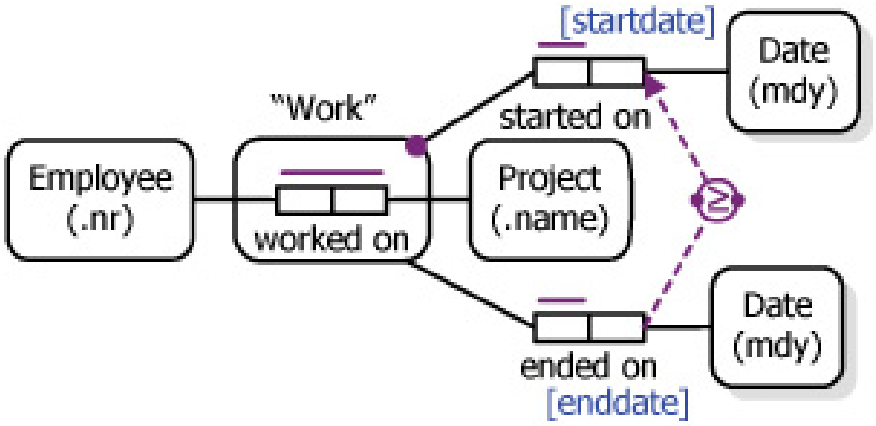
External uniqueness constraint involving an m:n fact type:



Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Rmap – Mapping Objectified Associations

- Initially treat the objectified association as a “black box” (i.e. remove its identification scheme)
- Group fact types in the usual way
- Unpack ”the black box” into its component attributes



⇨ *Work* (empNr, projectName, startdate, [enddate])[1]

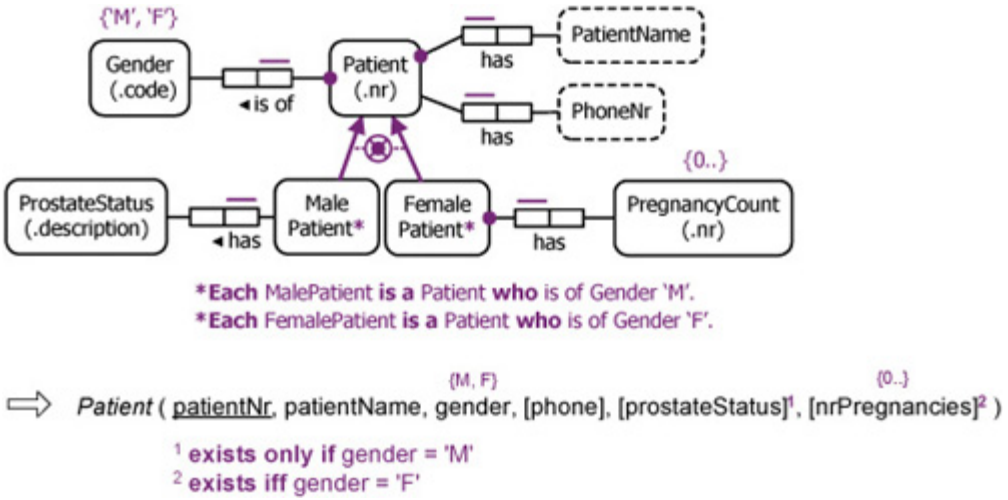
[1] enddate >= startdate

Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

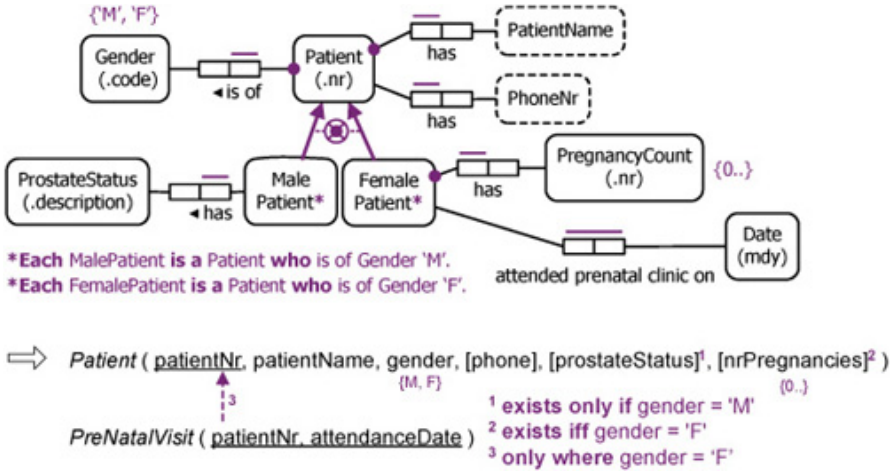
Rmap – Mapping Subtypes

- Subtyping mapping:
  - **Absorption**: absorb the subtypes back into the (top) supertype (giving qualified optional roles), group the fact types as usual, and then add the subtyping constraints as textual qualifications
  - Separation: separate tables for facts with subtype specific functional roles
  - Partition: may be used when the subtypes form a partition of their supertype

Subtype constraints on functional roles map to qualified optionals:



Nonfunctional fact type of subtypes map to separate tables:



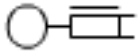
Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Basic Rmap procedure

STI · INNSBRUCK

0 Absorb subtypes into their top supertype.
Mentally erase all explicit preferred identification schemes,
treating compositely identified object types as "black boxes".

1 Map each fact type with a compound UC  to a separate table.

2 Fact types with functional roles attached to the same object type 
are grouped into the same table, keyed on the object type's identifier.
Map 1:1 cases to a single table, generally favoring fewer nulls.

3 Map each independent object type with no functional roles to a separate table.

4 Unpack each "black box column" into its component attributes.

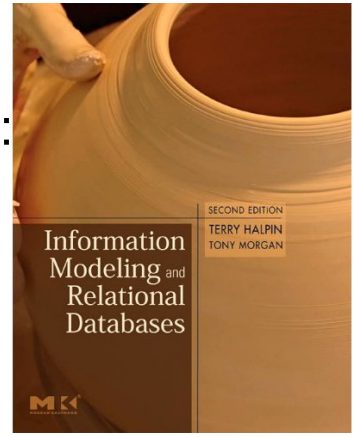
5 Map all other constraints and derivation rules.
Subtype constraints on functional roles map to qualified optional columns,
and those on nonfunctional roles map to qualified subset constraints.
Nonfunctional roles of independent object types map to column sequences
that reference the independent table.

Material on this slide based on *Ch 11.3* in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases*, Second Edition

Resources

STI · INNSBRUCK

- Chapter 11 in Halpin, T. & Morgan, T. 2008, *Information Modeling and Relational Databases, Second Edition* (ISBN: 978-0-12-373568-3), published by Morgan Kaufmann Publishers.



Next lecture

STI · INNSBRUCK

#	Title	Date
1	Introduction	07.10.2015
2	General concepts	21.10.2015
3	ORM modeling	04.11.2015
4	Relational modeling	18.11.2015
5	**ER modeling**	**02.12.2015**
6	OO modeling	16.12.2015
7	Services and process modeling	13.01.2016
10	Exam	27.01.2016

Questions?

STI · INNSBRUCK