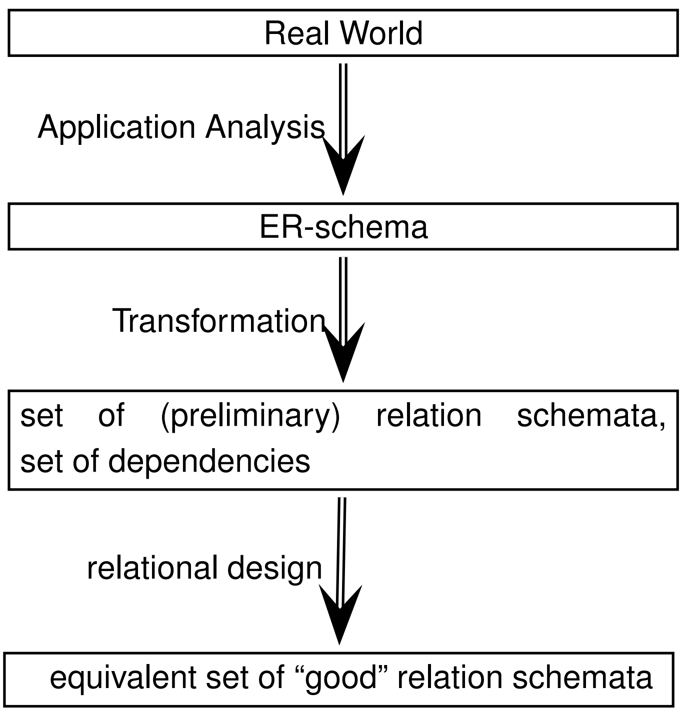
Chapter 7
Design Theory of the Relational Model

Goal: a relational schema that suitably represents an excerpt of the real world.

- Real world implies integrity constraints (we have seen e.g. keys and referential integrity as *relational concepts*)
- Base of such concepts: *data dependencies*
- Representation must cope with these dependencies (from this design, keys are obtained, and referential integrity constraints).

DESIGN STEPS



The more exact the ER model, the better the preliminary relational schema.

MOTIVATION

Example 7.1
Consider the following situation: a supplier has contracts with several customers to deliver products regularly. For each product, the number of delivered items and the price is relevant.

Pizza-Service				
Name	*Address*	*Product*	*Number*	*Price*
Meier	*Göttingen*	*Pizza*	*10*	*5.00*
Meier	*Göttingen*	*Lasagne*	*15*	*6.00*
Meier	*Göttingen*	*Salad*	*20*	*3.00*
Müller	*Kassel*	*Pizza*	*12*	*5.00*
Müller	*Kassel*	*Salad*	*15*	*3.00*

- *Redundancy*
- *caused problems:*

  (1) *anomalies when updating or inserting (potential inconsistencies),*

  (3) *anomalies when deleting (delete Meier $\rightarrow$ information about price of Lasagne is lost)*

Example 7.1 (Continued)
Refined Schema:

Customer	
Name	*Address*
Meier	*Göttingen*
Müller	*Kassel*

Product	
Product	*Price*
Pizza	*5.00*
Lasagne	*6.00*
Salad	*3.00*

Shipment′		
Name	*Product*	*Number*
Meier	*Pizza*	*10*
Meier	*Lasagne*	*15*
Meier	*Salad*	*20*
Müller	*Pizza*	*12*
Müller	*Salad*	*15*

is the refined schema "better"?

- *is it equivalent?*
- *anomalies removed?* □

Required Notions

1. Analysis of relevant dependencies
2. criterion when to decompose a relation schema (and when a decomposition is equivalent) (based on (1))
3. measure for "quality" of a schema
   (in terms of (1))

7.1 Functional Dependencies

- Data dependencies that describe a **functional** relationship.

Let $\bar{V}$ a set of attributes and $r \in \text{Rel}(\bar{V})$, $\bar{X}, \bar{Y} \subseteq \bar{V}$.
r satisfies the **functional dependency (FD)** $\bar{X} \to \bar{Y}$ if for all $t, s \in r$,

$$t[\bar{X}] = s[\bar{X}] \Rightarrow t[\bar{Y}] = s[\bar{Y}] \ .$$

For $\bar{Y} \subseteq \bar{X}$, $\bar{X} \to \bar{Y}$ is a **trivial** dependency (satisfied by every relation $r \in \text{Rel}(\bar{V})$).

Refined Definition of "Relation Schema"

A **relation schema** $R(\bar{X}, \Sigma_{\bar{X}})$ consists of a name (here, R) and a finite set $\bar{X} = \{A_1, \ldots, A_m\}$, $m \geq 1$ of attributes:

- $\bar{X}$ is the **format** of the schema.
- $\Sigma_{\bar{X}}$ is a set of functional dependencies over $\bar{X}$.

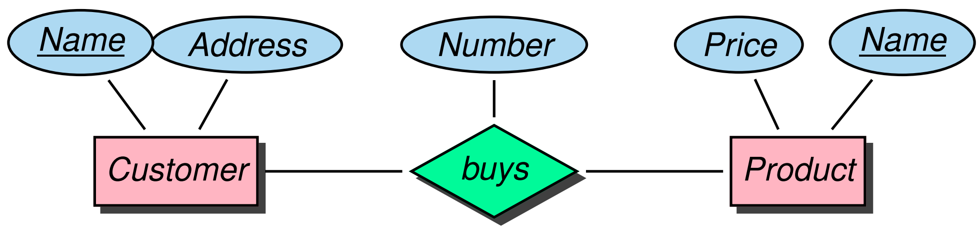
A relation $r \in \text{Rel}(\bar{X})$ is an **instance** of R if it satisfies all dependencies in $\Sigma_{\bar{X}}$.
The set of all instances of R is denoted by $\text{Sat}(\bar{X}, \Sigma_{\bar{X}})$.

Example 7.2
Consider again Example 7.1.

The given instance is in $Sat(\bar{X}, \Sigma_{\bar{X}})$ for the following set $\Sigma_{\bar{X}}$ of FDs:

$$\textit{Name} \rightarrow \textit{Address}$$
$$\textit{Product} \rightarrow \textit{Price}$$
$$\textit{(Name, Product)} \rightarrow \textit{Number}$$

"Intuitive" ER-model of the problem:



□

7.1.1 Decomposition Based on Functional Dependencies

- Does a "good" ER-model already guarantee all desirable properties of the relational model?

  NO

  (at least not completely - The more exact the ER model, the better the preliminary relational schema)

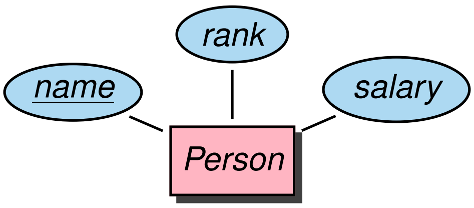
- is a formal dependency analysis necessary?

  YES

- theory: based on *normal forms* of relational schemata

Analysis of Entity Sets

Example 7.3 (FDs of entity attributes)
Consider a staff database in a university. Persons (professors and lecturers) have names, ranks, and salaries.



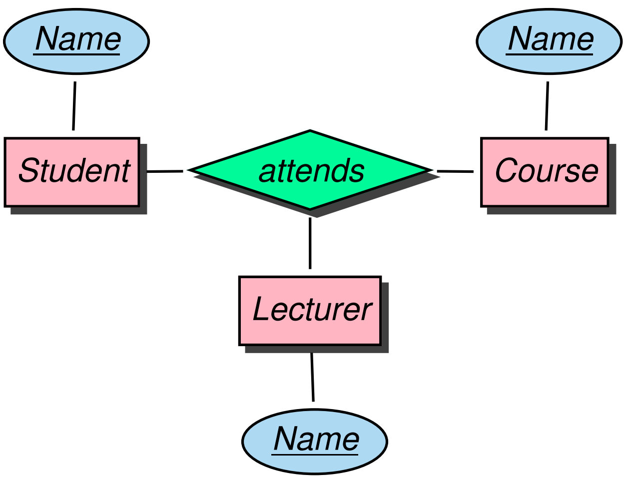
Person		
Name	Rank	Salary
G	full prof.	5000
T	full prof.	5000
S	associate prof.	4000
W	assistant	3000
P	assistant	3000

There is a functional dependency Rank → Salary.

Refined schema: *Person(Name, Rank)*
SalaryTable(Rank, Salary) □

Analysis of Relationship Sets

Example 7.4 (FDs of ternary relationships)
Students attend courses that are given by lecturers.



attends		
Student	Course	Lecturer
Stud1	Telematics	Ho
Stud2	Telematics	Ho
Stud2	Mobile Comm	Ho
Stud3	Mobile Comm	Ho
Stud3	Databases	WM
Stud4	Databases	WM
Stud1	Databases	WM

There is a functional dependency Course → Lecturer.

Refined schema: *reads(Course, Lecturer)*
attends'(Student, Course) □

7.1.2 Functional Dependency Theory

Let $R(\bar{V}, \mathcal{F})$ a relation schema where $\bar{X}, \bar{Y} \subseteq \bar{V}$, and $\mathcal{F}$ is a set of functional dependencies over $\bar{V}$.

Definition 7.1

- *$\mathcal{F}$ **implies** a functional dependency $\bar{X} \to \bar{Y}$, written as $\mathcal{F} \models \bar{X} \to \bar{Y}$, if and only if every relation $r \in$ Sat$(\bar{V}, \mathcal{F})$ satisfies $\bar{X} \to \bar{Y}$.*
- *$\mathcal{F}^+ = \{\bar{X} \to \bar{Y} \mid \mathcal{F} \models \bar{X} \to \bar{Y}\}$ is the **closure** of $\mathcal{F}$.* □

Definition 7.2

*Let $\bar{V} = \{A_1 \ldots A_n\}$. $\bar{X}$ is a **key** of $\bar{V}$ (wrt. $\mathcal{F}$) if and only if*

- $\bar{X} \to A_1 \ldots A_n \in \mathcal{F}^+$,
- $\bar{Y} \subsetneq \bar{X} \Rightarrow \bar{Y} \to A_1 \ldots A_n \notin \mathcal{F}^+$.

For a key $\bar{X}$, each $\bar{Y} \supseteq \bar{X}$ is a superkey. □

For an attribute A such that $A \in \bar{X}$ for any key $\bar{X}$, A is a **key attribute**. If there is no key $\bar{X}$ such that $A \in \bar{X}$, then A is a **non-key attribute**.

Closure of FDs

Problem: How to decide whether $\bar{X} \to \bar{Y} \in \mathcal{F}^+$? (Membership Test)

The test is based on the **Armstrong-Axioms**:

Let $\mathcal{F}$ a set of FDs over $\bar{V}$ and $r \in$ Sat$(\bar{V}, \mathcal{F})$.

(A1) Reflexivity: If $\bar{Y} \subseteq \bar{X} \subseteq \bar{V}$, then r satisfies $\bar{X} \to \bar{Y}$.

(A2) Augmentation: If $\bar{X} \to \bar{Y} \in \mathcal{F}$ and $\bar{Z} \subseteq \bar{V}$, then r satisfies $\overline{XZ} \to \overline{YZ}$.

(A3) Transitivity: If $\bar{X} \to \bar{Y}$ and $\bar{Y} \to \bar{Z} \in \mathcal{F}$, then r satisfies $\bar{X} \to \bar{Z}$.

The Armstrong-Axioms can be used as inference rules for FDs.

Theorem 7.1

*The Armstrong-Axioms are **correct**, i.e., all derived FDs are in $\mathcal{F}^+$, and they are **complete**, i.e., all FDs in $\mathcal{F}^+$ can be derived.* □

CLOSURE OF FDS (CONT'D)

Armstrong Axioms can especially be used for searching which attributes depend on a given $\bar{X} \subseteq V$.

Definition 7.3
For $\bar{X} \subseteq \bar{V}$, $\bar{X}^+$ is the set of all $A \in \bar{V}$ such that $\bar{X} \to A$ can be derived by the Armstrong axioms. $\bar{X}^+$ is called the ***(Attribute-)closure*** *of $\bar{X}$ (wrt. $\mathcal{F}$).* □

Exercise 7.1
Consider a relation schema $R(\bar{V}, \mathcal{F})$ such that $\bar{K}$ is a key. What is $\bar{K}^+$? □

Proof of Theorem 7.1: correctness is obvious.
Completeness: it has to be shown that if $\bar{X} \to \bar{Y} \in \mathcal{F}^+$, then $\bar{X} \to \bar{Y}$ can be derived by (A1)–(A3) from $\mathcal{F}$.

It will be shown: if $\bar{X} \to \bar{Y}$ is not derivable by (A1)–(A3), then $\bar{X} \to \bar{Y} \notin \mathcal{F}^+$, i.e., there is an $r \in Sat(\bar{V}, \mathcal{F})$ that does not satisfy $\bar{X} \to \bar{Y}$.

Assume $\bar{X} \to \bar{Y}$ cannot be derived. Consider a relation r as below:

$$\begin{array}{cc} 1\ \ 1\ \ \ldots\ \ 1 & 1\ \ 1\ \ \ldots\ \ 1 \\ \underbrace{1\ \ 1\ \ \ldots\ \ 1}_{\text{attributes in } \bar{X}^+} & \underbrace{0\ \ 0\ \ \ldots\ \ 0}_{\text{all other attributes}} \end{array}$$

(i) First it will be shown that r satisfies $\mathcal{F}$:
Assume that there is a $\bar{Z} \to \bar{W} \in \mathcal{F}$, such that r does not satisfy $\bar{Z} \to \bar{W}$. This is only possible if $\bar{Z} \subseteq \bar{X}^+$ and $W \not\subseteq \bar{X}^+$. Since $\bar{Z} \subseteq \bar{X}^+$, there is $\bar{X} \to \bar{Z}$ and $\bar{Z} \to W$, and thus $W \subseteq \bar{X}^+$, a contradiction.

(ii) Next, it will be shown that r does not satisfy $\bar{X} \to \bar{Y}$:
For any $\bar{X} \to \bar{Y}$ that is satisfied by r, $\bar{Y} \subseteq \bar{X}^+$. This would mean that $\bar{X} \to \bar{Y}$ can be derived from (A1)–(A3).

Membership Problem

Check whether $\bar{X} \rightarrow \bar{Y} \in \mathcal{F}^+$?

Variant 1 :
Compute $\mathcal{F}^+$ from $\mathcal{F}$ using (A1)–(A3) until either $\bar{X} \rightarrow \bar{Y}$ is derived, or the process stops. Then, $\mathcal{F}^+$, and $\bar{X} \rightarrow \bar{Y} \notin \mathcal{F}^+$.

This algorithm is not efficient, since it has (systematically applied) at least the time complexity $O(2^{||\mathcal{F}||})$.

Example 7.5
Consider $\bar{V} = \{A, B_1, \ldots, B_n, C, D\}$ with $\mathcal{F} = \{A \rightarrow B_1, \ldots, A \rightarrow B_n\}$. Then, $A \rightarrow \bar{Y} \in \mathcal{F}^+$ for all $\bar{Y} \subseteq \{B_1, \ldots, B_n\}$. Thus, computation of $\mathcal{F}$ needs to compute 2^n items (before a negative answer for any other FD – e.g. the question whether $C \rightarrow D$ holds – can be stated). □

Membership Problem (Cont'd)

Variant 2 :
Goal-oriented approach for $\bar{X} \rightarrow \bar{Y} \in \mathcal{F}^+$:

Compute $\bar{X}^+$ and check if $\bar{Y} \subseteq \bar{X}^+$.

- start with $X \rightarrow X$ (A1 - Reflexivity)
- (A2) allows $\bar{X} \rightarrow \bar{Y} \in \mathcal{F} \Rightarrow \overline{XX} \rightarrow \overline{XY} \in \mathcal{F}^+$ which is equivalent to $\bar{X} \rightarrow \overline{XY} \in \mathcal{F}^+$
- for any $\bar{Z} \supset \bar{X}$ and $\bar{X} \rightarrow \overline{XY} \in \mathcal{F}^+$, (A2) allows to conclude $\bar{Z} \rightarrow \overline{ZY}$ (A2*)
- "collect" $\bar{X}^+$ in this way: derive $\bar{X} \rightarrow \overline{XY_1}$, then $\overline{XY_1} \rightarrow \overline{XY_2}$ by (A2*) and apply (A3 - transitivity) to them,
- until $\bar{X} \rightarrow \bar{Z} \in \mathcal{F}^+$ for $\bar{Y} \subset \bar{Z}$, then derive $\bar{X} \rightarrow \bar{Y} \in \mathcal{F}^+$ by (A1).

Example 7.6
$\mathcal{F} = \{AB \to E, BE \to I, E \to G, GI \to H\}$, check if $AB \to GH \in \mathcal{F}^+$?

	$X \to Y \in \mathcal{F}$	*and derive ...*
(A1)	$AB \to AB$	
(A2)*	$AB \to E$	$AB \to ABE$
(A2)*	$BE \to I$	$ABE \to ABEI$
(A2)*	$E \to G$	$ABEI \to ABEIG$
(A2)*	$GI \to H$	$ABEIG \to ABEIGH$
(A3) transitivity:		$AB \to ABEIGH$
final step with (A1):		$AB \to GH$

□

Membership Problem (Cont'd)

- consider each (A2*) + (A3) step as one:

$\bar{X}^+$-Algorithm:

```
result := X̄;      /* (A1) */
WHILE (changes to result) DO
  FOR each W̄ → Z̄ ∈ F DO      /* (A2*) + (A3) */
     IF (W̄ ⊆ result ) THEN result := result ∪ Z̄ ;
  end;
check if Ȳ ⊆ result      /* (A1) */;
```

Theorem 7.2
The $\bar{X}^+$-algorithm computes $\bar{X}^+$ and terminates. Its time complexity is $O((|\mathcal{F}| \cdot |V|)^2)$.
There is an optimized variant in $O(|\mathcal{F}| \cdot |V|)$. □

Example 7.7
Apply the $\bar{X}^+$-algorithm to Example 7.6 (same steps). □

AN EQUIVALENT SET OF RULES

Lemma 7.1
Consider a relation schema $R(\bar{V},\mathcal{F})$ such that $A \in \bar{V}$ and $\bar{X},\bar{Y},\bar{Z},\bar{W} \subseteq \bar{V}$, and $\mathcal{F}$ is a set of functional dependencies over $\bar{V}$, and $r \in$ Sat$(\bar{V},\mathcal{F})$. Then:

- **(A4)** *Union: If $\bar{X} \to \bar{Y}$ and $\bar{X} \to \bar{Z} \in \mathcal{F}$, then r satisfies $\bar{X} \to \overline{YZ}$.*
- **(A5)** *Pseudo-transitivity: If $\bar{X} \to \bar{Y}$ and $\overline{WY} \to \bar{Z} \in \mathcal{F}$, then r satisfies $\overline{XW} \to \bar{Z}$.*
- **(A6)** *Decomposition: If $\bar{X} \to \bar{Y} \in \mathcal{F}$ and $\bar{Z} \subseteq \bar{Y}$, then r satisfies $\bar{X} \to \bar{Z}$.*
- **(A7)** *Reflexivity: r satisfies $\bar{X} \to \bar{X}$*
- **(A8)** *Accumulation: If $\bar{X} \to \overline{YZ}$ and $\bar{Z} \to \overline{AW} \in \mathcal{F}$, then r satisfies $\bar{X} \to \overline{YZA}$.*

□

Lemma 7.2
The rule sets {(A1), (A2), (A3)} and {(A6), (A7), (A8)} are equivalent, i.e., for given $\mathcal{F}$, the same set of FDs can be derived. □

- (A8) covers the combination of (A2*) and (A3) (consider $\bar{W} = \emptyset$).

Example 7.8
$\mathcal{F} = \{AB \to E, BE \to I, E \to G, GI \to H\}$, check if $AB \to GH \in \mathcal{F}^+$?

Derivation by (A7)–(A8)		Intermediate result $\bar{X}_i$ of the $\bar{X}^+$-algorithm
(A7)	$AB \to AB$	$\bar{X}_0 = \{A, B\}$
(A8)	$[\,AB \to E\,]$	
	$AB \to ABE$	$\bar{X}_1 = \{A, B, E\}$
(A8)	$[\,BE \to I\,]$	
	$AB \to ABEI$	$\bar{X}_2 = \{A, B, E, I\}$
(A8)	$[\,E \to G\,]$	
	$AB \to ABEIG$	$\bar{X}_3 = \{A, B, E, I, G\}$
(A8)	$[\,GI \to H\,]$	
	$AB \to ABEIGH$	$\bar{X}_4 = \{A, B, E, I, G, H\}$
final step with (A6):		
(A6)	$AB \to GH$	

□

Determining a Key

Consider a relation schema $R = (\bar{V}, \mathcal{F})$.

- The $\bar{X}^+$-algorithm allows for determining a key of R in time $O(|\mathcal{F}|\ |\bar{V}|^2)$:
  Start with the superkey $\bar{V}$ and try to delete attributes as long as the closure of the remaining attributes is still the whole $\bar{V}$. If no more attributes can be deleted, a key is found.
- In the general case, it is not possible to determine *all* keys of a relation schema efficiently. Note that the problem "is there a key with at most k attributes?" is NP-complete.

Aside: Unique Keys

Theorem 7.3
Let $\mathcal{F} = \{\bar{X}_1 \to \bar{Y}_1, \ldots, \bar{X}_p \to \bar{Y}_p\}$.
Let $\bar{Z}_i = \bar{Y}_i \setminus \bar{X}_i$ *for* $1 \leq i \leq p$.

$R(\bar{V})$ *has a unique key if and only if* $\bar{V} \setminus (\bar{Z}_1 \cup \ldots \cup \bar{Z}_p)$ *is a superkey.*
(note that $\bar{K}$ *is a superkey if* $\bar{K}^+ = \bar{V}$*).*
(Proof: next slide) □

Note:

- $\bar{Z}_1 \cup \ldots \cup \bar{Z}_p$ contains those attributes that are fd from any other attribute.
- $\bar{V} \setminus (\bar{Z}_1 \cup \ldots \cup \bar{Z}_p)$ contains those attributes that are not fd from any other attribute.
- $\bar{V} \setminus (\bar{Z}_1 \cup \ldots \cup \bar{Z}_p)$ is subset of all keys of a relation.

Example 7.9
Consider the relation Country(name,code,population, area) *with FDs*
name$\to$ code,population,area *and* code$\to$ name,population,area.
Here, name *and* code *are keys.*
$\bar{V} \setminus (\ldots) = \emptyset$ □

ASIDE: UNIQUE KEYS (CONT'D)

Proof of Theorem 7.3:

"⇒" Assume $\bar{K}$ to be the unique key of R. Then, $\bar{K}$ is contained in every superkey. For each $1 \leq i \leq p$, $\bar{V} \setminus \bar{Z}_i$ is a superkey (since $\bar{Z}_i$ is determined by $\bar{X}_i$).

Thus, $\bar{K} \subseteq \cap_{i=1}^{p}(\bar{V} \setminus \bar{Z}_i)$. The right side is equivalent to $\bar{V} \setminus (\bar{Z}_1 \cup \ldots \cup \bar{Z}_p)$. Thus, $\bar{V} \setminus (\bar{Z}_1 \cup \ldots \cup \bar{Z}_p)$ is a superkey (of $\bar{K}$).

"⇐" Assume $\bar{K} = \bar{V} \setminus (\bar{Z}_1 \cup \ldots \cup \bar{Z}_p)$ a superkey. It will be shown that $\bar{K}$ is contained in every superkey, and thus it is the only key. Suppose a superkey $\bar{L}$ such that there is an attribute $A \in \bar{K} \setminus \bar{L}$. Then, $A \notin \bar{L}^+$ (since it is not in any of the $\bar{Z}_i$). Thus, L is not a superkey (since $\bar{L}^+ \subsetneq \bar{V}$) – contradiction.

SETS OF FDS

Consider sets $\mathcal{F}, \mathcal{G}$ of functional dependencies. $\mathcal{F}, \mathcal{G}$ are **equivalent** if and only if $\mathcal{F}^+ = \mathcal{G}^+$.

Definition 7.4
*$\mathcal{F}$ is **minimal** if and only if*

1. *For every $\bar{X} \to \bar{Y} \in \mathcal{F}$, $\bar{Y}$ consists of a single attribute,*
2. *For every $\bar{X} \to A \in \mathcal{F}$, $\mathcal{F} \setminus \{\bar{X} \to A\}$ is not equivalent to $\mathcal{F}$,*
3. *If $\bar{X} \to A \in \mathcal{F}$ and $\bar{Z} \subset \bar{X}$, then $\mathcal{F} \setminus \{\bar{X} \to A\} \cup \{\bar{Z} \to A\}$ is not equivalent to $\mathcal{F}$.* □

Theorem 7.4
For each set $\mathcal{F}$ of functional dependencies, there is an equivalent minimal set $\mathcal{F}^{min}$ of functional dependencies.
(Note: $\mathcal{F}^{min}$ is not necessarily unique). □

Example 7.10
Consider again Example 7.9:
*{*name→ {code}, name→ {population}, name→ {area}, code→{name}*}*
*and {*code→ {name}, code→ {population}, code→ {area}, name→{code}*}*
are minimal. □

MINIMAL SETS OF FDS

- $\mathcal{F}^{min}$ can be computed by the $\bar{X}^+$-algorithm (without computing $\mathcal{F}^+$) in polynomial time.

Consider a schema $R(\bar{V}, \mathcal{F})$ with $|\bar{V}| = n$ and $|\mathcal{F}| = f$.

1. Decompose all $X \to Y \in \mathcal{F}$ such that each right side consists of a single attribute; get $\mathcal{F}'$ with $|\mathcal{F}'| \leq nf$ in $O(f \cdot n)$ steps.
2. Delete all $\varphi \in \mathcal{F}'$ that follow from the others (iteratively), using the X^+ algorithm. Each application of X^+ requires $O(f \cdot n)$ steps, thus, altogether $O(f^2 \cdot n^2)$.
3. Delete in each remaining FD $\{x_1 \ldots, x_n\} \to y$ stepwise as many attributes on the left side as possible. For each step, check, whether y is still in the remaining $\{x_1 \ldots, x_k\}^+$. The X^+-algorithm is applied $|\mathcal{F}'| \cdot n = O(f \cdot n^2)$ times, thus, this step is in $O(f^2 \cdot n^3)$.
4. The algorithm is in $O(f^2 \cdot n^3)$, i.e., polynomial.

7.2 Decomposition of Relation Schemata

In Example 7.1 (Slide 328), a relation has been *decomposed* for yielding a better behavior.

Definition 7.5

- *Let $\bar{V}$ a set of attributes. Then, $\rho = \{\bar{X}_1, \ldots, \bar{X}_n\}$ s.t. $\bar{X}_1 \cup \ldots \cup \bar{X}_n = \bar{V}$ and for each i, $\bar{X}_i \subseteq \bar{V}$ is a **decomposition** of $\bar{V}$.* □

Example 7.11

Consider again Example 7.1. There, $\bar{V}$ = {Name, Address, Product, Number, Price}.

E.g., ρ = {{Name, Address}, {Product, Price}, {Name, Product, Number}}. is a decomposition. □

Lemma 7.3

Consider a relation $r \in$ Rel$(\bar{V})$ and a decomposition $\rho = \{\bar{X}_1, \ldots, \bar{X}_k\}$ of $\bar{V}$.

Then,

$$r \subseteq \pi[\bar{X}_1](r) \bowtie \ldots \bowtie \pi[\bar{X}_k](r)\ .$$ □

PROPERTIES OF DECOMPOSITIONS

Losslessness: The complete tuples must be reconstructable by joining the decomposed relations without getting additional tuples that have not been there originally.

Example 7.12
Consider again Example 7.4, now with a decomposition into hears(Student,Lecturer) *and* attends'(Student, Course).

Then, the join hears $\bowtie$ attends' *yields a tuple (*DStud1,Databases,Ho*).* □

Definition 7.6
Consider a relation schema $R(\bar{V},\mathcal{F})$ and a decomposition $\rho = \{\bar{X}_1,\ldots,\bar{X}_n\}$ of R.

*ρ is **lossless** if and only if for every relation $r \in$ Sat$(\bar{V},\mathcal{F})$,*

$$r = \pi[\bar{X}_1](r) \bowtie \ldots \bowtie \pi[\bar{X}_k](r) \ .$$ □

PROPERTIES OF DECOMPOSITIONS (CONT'D)

dependency-preservation: the dependencies can be tested using the decomposed tables only, i.e., without having to recompute the join.

Definition 7.7
Consider a relation schema $R(\bar{V},\mathcal{F})$ and a decomposition $\rho = \{\bar{X}_1,\ldots,\bar{X}_n\}$ of R.

$\pi[Z](\mathcal{F}) = \{X \to Y \in \mathcal{F}^+ \mid XY \subseteq Z\}$ is the projection of $\mathcal{F}$ to Z.

*ρ is **dependency-preserving** if and only if for all i,*

$$\bigcup_{i=1}^{n} \pi[\bar{X}_i](\mathcal{F}) \equiv \mathcal{F} \ .$$ □

Dependency-preservation means that FDs can be distributed over the decomposition without losing anything:
If the projections of $\mathcal{F}^+$ are asserted, the (joined) database contents satisfies $\mathcal{F}$.

We will first discuss losslessness.

7.2.1 Lossless Decompositions

- The problem is not to lose tuples by (wrong) decompositions, but to lose "information" about relationships.

Example 7.13
Consider again Examples 7.4 and 7.12.

1. $attends = \underbrace{\pi[Course, Lecturer](attends)}_{reads} \bowtie \underbrace{\pi[Student, Course](attends)}_{attends'}$
2. $attends \subsetneq \underbrace{\pi[Student, Lecturer](attends)}_{hears} \bowtie \underbrace{\pi[Student, Course](attends)}_{attends'}$
   (DStud1,Databases,Ho) $\in$ hears $\bowtie$ attends'. □

Test for Losslessness (Chase-Algorithm for FDs)

Input: a relation schema $R(\bar{V}, \mathcal{F})$, where $\bar{V} = \{A_1, \ldots, A_n\}$, $\rho = \{\bar{X}_1, \ldots, \bar{X}_k\}$.

Algorithm: (Aho, Beeri, Ullman, TODS 1979)

Idea: take a tuple $(a_1, \ldots, a_n)$, decompose it according to ρ. Create a "test table" that represents the requirements of a tuple $(a_1, \ldots, a_n)$ in the re-join of the decomposed tables. Add the knowledge from the FDs about the attribute values of this tuple. The goal is to show that this tuple must have been already present in the original table.

Construct a table T with n columns and k rows.
Column j stands for A_j, row i for $\bar{X}_i$ as follows:

- $T_{(i,j)} = a_j$ if $A_j \in \bar{X}_i$,
- otherwise $T_{(i,j)} = b_{ij}$ ("any value").

(see next slide)

Chase-Algorithm for FDs (Cont'd)

As long as T changes, do the following:

Consider a FD $\bar{X} \to \bar{Y} \in \mathcal{F}$. If there are rows $z_1, z_2 \in T$ which coincide for all $\bar{X}$-columns, but not in all $\bar{Y}$-columns, then make their $\bar{Y}$-values the same:

- For each $\bar{Y}$-column j:
- if one of the symbols is a_j, then replace every occurrence of the other symbol globally by a_j.
- if both symbols are of the form b_{ij}, then replace arbitrarily one of them globally by the other.

Note: The algorithm corresponds to *enforcing* the FDs.
(since they are known to hold in T, this constrains the occurrences of other values)

Result: ρ is lossless if and only if $(a_1, \ldots, a_n) \in T$.

Example 7.14 (Chase)
$\bar{V} = ABCDE$, $\rho = (AD, AB, BE, CDE, AE)$;
$\mathcal{F} = \{A \to B, B \to D, DE \to C, E \to A\}$

	A	B	C	D	E
from AD:	a_1	b_{12}	b_{13}	a_4	b_{15}
from AB:	a_1	a_2	b_{23}	b_{24}	b_{25}
from BE:	b_{31}	a_2	b_{33}	b_{34}	a_5
from CDE:	b_{41}	b_{42}	a_3	a_4	a_5
from AE:	a_1	b_{52}	b_{53}	b_{54}	a_5

$\xrightarrow{\text{chase}}$

A	B	C	D	E
a_1	a_2	b_{13}	a_4	b_{15}
a_1	a_2	b_{23}	a_4	b_{25}
$\underline{a_1}$	$\underline{a_2}$	$\underline{a_3}$	$\underline{a_4}$	$\underline{a_5}$
a_1	b_{42}	a_3	a_4	a_5
a_1	b_{52}	b_{53}	b_{54}	a_5

The process is finished when the following table is reached:

A	B	C	D	E
a_1	a_2	a_3	a_4	b_{15}
a_1	a_2	a_3	a_4	b_{25}
a_1	a_2	a_3	a_4	a_5
a_1	a_2	a_3	a_4	a_5
a_1	a_2	a_3	a_4	a_5

Note that only for columns that do not occur on the right side of a FD, the bs remain.

□

Theorem 7.5
The above algorithm for testing losslessness is correct. □

Proof:

Notation:

- for a decomposition $\rho = \{\bar{X}_1, \ldots, \bar{X}_k\}$ of $\bar{V}$ and a relation r, the re-join of the decomposed tables is denoted by $m_\rho(r) = \bowtie_{i=1}^{k} \pi[\bar{X}_i](r)$.
- T_0 and T^* denote the table before and after execution of the algorithm.

The algorithm terminates since the number of different symbols decreases with every step.

(A) It has to be shown that if ρ is lossless, $(a_1, \ldots, a_n) \in T^*$.

Due to the construction of T_0, each $\pi[\bar{X}_i](T_0)$ contains a row that consists only of a_is. Thus, $(a_1, \ldots, a_n) \in m_\rho(T_0)$.

This property is preserved by the chase steps, thus $(a_1, \ldots, a_n) \in m_\rho(T^*)$. The chase process also guarantees that $T^* \in$ Sat$(\bar{V}, \mathcal{F})$. From the assumption that ρ is lossless, $T^* = m_\rho(T^*)$ and $(a_1, \ldots, a_n) \in T^*$.

(B) (uses Relational Calculus)

It will be shown that if $(a_1, \ldots, a_n) \in T^*$, ρ is lossless.

Consider relations r over $R(\bar{V})$ (as structures). Consider the formula of the calculus

$$F_0 = (\exists b_{11}) \ldots (\exists b_{kn})(R(w_1) \wedge \ldots \wedge R(w_k))$$

where w_i is the i-th row of T_0 and all a_i and b_{jk}'s are interpreted as variables. The free variables in F_0 are $a_1, \ldots, a_n$. Note that every member $R(w_i)$ of the conjunction in F_0 corresponds to a projection to $\bar{X}_i$. Then,

$$m_\rho(r) = \text{answers}(F_0(a_1, \ldots a_n)) \ .$$

Consider only relations $r \in$ Sat$(\bar{V}, \mathcal{F})$. Since r satisfies $\mathcal{F}$,

$$F_0(a_1, \ldots a_n) \equiv_\mathcal{F} F_1(a_1, \ldots a_n) \equiv_\mathcal{F} \ldots \equiv F^*(a_1, \ldots a_n)$$

where each F_i corresponds to the table after i chase steps. For given r, the answer set to F^* is the same as the answer set to F_0.

Since $F^*(a_1, \ldots a_n)$ is of the form $(\exists b_{11}) \ldots (\exists b_{km})(R(a_1, \ldots, a_n) \wedge \ldots)$,
its answer set is a subset (or equal) of r.

Altogether, $m_\rho(r) \subseteq r$. Since $m_\rho(r) \supseteq r$ by Lemma 7.3, $m_\rho(r) = r$, i.e., ρ is lossless.

Corollary 7.1 (Decomposition into two relations)
Consider a set $\bar{V}$ of attributes, a set $\mathcal{F}$ of functional dependencies, and a decomposition $\rho = \{\bar{X}_1, \bar{X}_2\}$ of $\bar{V}$. ρ is lossless if and only if

$$(\bar{X}_1 \cap \bar{X}_2) \to (\bar{X}_1 \setminus \bar{X}_2) \in \mathcal{F}^+, \text{ or } (\bar{X}_1 \cap \bar{X}_2) \to (\bar{X}_2 \setminus \bar{X}_1) \in \mathcal{F}^+ \ . \qquad \square$$

Proof:
The table T for ρ is

	$\bar{X}_1 \cap \bar{X}_2$	$\bar{X}_1 \setminus \bar{X}_2$	$\bar{X}_2 \setminus \bar{X}_1$
$\bar{X}_1$	$a \ldots a$	$a \ldots a$	$b \ldots b$
$\bar{X}_2$	$a \ldots a$	$b \ldots b$	$a \ldots a$

1. Assume $(a_1, \ldots, a_n) \in T^*$. Consider an attribute A whose column contains a b. If the algorithm exchanges it by an a, then $A \in (\bar{X}_1 \cap \bar{X}_2)^+$. Due to the assumption that $(a_1, \ldots, a_n) \in T^*$, there is one line where this happens for all attributes – thus all these attributes are in $(\bar{X}_1 \cap \bar{X}_2)^+$.
2. Assume (w.l.o.g.) that $(\bar{X}_1 \cap \bar{X}_2) \to (\bar{X}_1 \setminus \bar{X}_2) \in \mathcal{F}^+$, i.e., $\bar{X}_1 \setminus \bar{X}_2 \subseteq (\bar{X}_1 \cap \bar{X}_2)^+$. Consider the steps for deriving this by the $\bar{X}^+$-algorithm. For each such step there is a corresponding chase-step. Thus, the chase replaces each b of an attribute in $\bar{X}_1 \setminus \bar{X}_2$ by an a, leading to $(a_1, \ldots, a_n) \in T^*$.

Example 7.15
Consider again Examples 7.4, 7.12 and 7.13 whith the schema

$$\textit{attends}((\textit{Student, Course, Lecturer}), \{\textit{Course} \to \textit{Lecturer}\})$$

- $\rho_1 = \{\{\textit{Course}, \textit{Lecturer}\}, \{\textit{Student}, \textit{Course}\}\}$ *is lossless.*
- $\rho_2 = \{\{\textit{Student}, \textit{Lecturer}\}, \{\textit{Student}, \textit{Course}\}\}$ *is not lossless.* $\square$

General conclusion for ternary relations:

- for any (useful) decomposition into two binary relations, there is one attribute A that is contained in both relations.
- the decomposition is lossless if at least one of the other attributes is functionally dependent only on A.

In the above example, the functional dependency *Course* $\to$ *Lecturer* which made the decomposition possible.

7.2.2 Dependency Preservation

Example 7.16

Consider again Examples 7.1 and 7.11 with the schema

$$\textit{Pizza-Service}(\ \{\textit{Name}, \textit{Address}, \textit{Product}, \textit{Number}, \textit{Price}\},\ \{\textit{Name} \to \textit{Address},\ \textit{Product} \to \textit{Price},\ (\textit{Name}, \textit{Product}) \to \textit{Number}\})$$

and the decomposition

$$\rho = \{\{\textit{Name}, \textit{Address}\},\ \{\textit{Product}, \textit{Price}\},\ \{\textit{Name}, \textit{Product}, \textit{Number}\}\}\ .$$

Recall that $\pi[Z](\mathcal{F}) = \{X \to Y \in \mathcal{F}^{+} \mid XY \subseteq Z\}$

$\pi[\textit{Name, Address}](\mathcal{F}) \supseteq \{\textit{Name} \to \textit{Address}\}$
$\pi[\textit{Product, Price}](\mathcal{F}) \supseteq \{\textit{Product} \to \textit{Price}\}$
$\pi[\textit{Name, Product, Number}](\mathcal{F}) \supseteq \{(\textit{Name,Product}) \to \textit{Number}$

So, all FD's immediately survive. □

Another, abstract Example

Example 7.17

$V = \{A, B, C, D\}$, $\rho = \{AB, BC\}$
$\mathcal{F} = \{A \to B, B \to C, C \to A\}$

ρ is dependency-preserving (check whether $C \to A$ is preserved).

Recall again that $\pi[Z](\mathcal{F}) = \{X \to Y \in \mathcal{F}^{+} \mid XY \subseteq Z\}$

($\mathcal{F}^{+}$ contains $A \to ABC$, $B \to ABC$, $C \to ABC$)

$\pi[AB](\mathcal{F}) \supseteq \{A \to B, B \to A\}$
$\pi[BC](\mathcal{F}) \supseteq \{B \to C, C \to B\}$
$C \to A \in (\pi[AB](\mathcal{F}) \cup \pi[BC](\mathcal{F}))^{+}$ □

DEPENDENCY PRESERVATION

There are lossless decompositions that are not dependency-preserving:

Example 7.18
Consider $R = (\bar{V}, \mathcal{F})$, where $\bar{V} = \{$City, Address, Zip$\}$, and
$\mathcal{F} = \{($City,Address$) \to$ Zip, Zip $\to$ City$\}$.

The decomposition R_1(Address, Zip) *and* R_2(City, Zip) *is lossless since* $(R_1 \cap R_2) \to (R_2 \setminus R_1) \in \mathcal{F}$*, but is not dependency-preserving.*

*(note that the keys of R are (*Address, Zip*) and (*City, Address*).)*

R	City	Address	Zip
	FR	Herdern	79106
	FR	Flughafen	79110
	FR	Mooswald	79110
	S	Flughafen	70629

R_1	Address	Zip
	Herdern	79106
	Flughafen	79110
	Mooswald	79110
	Flughafen	70629

R_2	City	Zip
	FR	79106
	FR	79110
	S	70629

*Insert (*FR,Herdern,79100*) and check the FDs:*
The original FD (City,Address) $\to$ Zip is not satisfied. □

... and now to a systematic characterization:

- some properties have been identified that should hold for a decomposition,
- algorithms have been giving for testing them;
- is it possible to express properties of such decompositions based on schema information?
- how to find such decompositions?

7.3 Normal Forms based on FDs

Task:

Consider a schema $R = (\bar{V}, \mathcal{F})$. Find a decomposition $\rho = (\bar{X}_1, \ldots, \bar{X}_n)$ of R such that

1. each $R_i = (\bar{X}_i, \pi[\bar{X}_i](\mathcal{F}))$, $1 \leq i \leq n$ is in some normal form,
2. ρ is lossless and (if possible) dependency-preserving,
3. n is minimal.

Non-normalized Data

Nested Relations:

Nested_Languages			
Code	**Name**	**Languages**	
D	Germany	German	100
CH	Switzerland	German	65
		French	18
		Italian	12
⋮	⋮	⋮	

Non-atomic values:

Products		
Code	**GDP**	**Products**
D	1452200	steel, coal, chemicals, machinery, vehicles
CH	158500	machinery, chemicals, watches
⋮	⋮	⋮

1ST NORMAL FORM (1NF)

Definition 7.8
A relation schema is in 1NF if and only if its attribute domains are atomic. □

Non-normalized relations are transformed into 1NF by expanding groups.

Note that redundancy arises (expressed by functional dependencies).

Example 7.19

Languages			
Code	**Name**	**Language**	**Percent**
D	*Germany*	*German*	*100*
CH	*Switzerland*	*German*	*65*
CH	*Switzerland*	*French*	*18*
CH	*Switzerland*	*Italian*	*12*
⋮	⋮	⋮	⋮

$$\mathcal{F} = \{\textit{Code} \to \textit{Name},\ \textit{Name} \to \textit{Code},\ (\textit{Code, Language}) \to \textit{Percent},\ (\textit{Name, Language}) \to \textit{Percent}\}$$ □

Example 7.20

Economy		
Code	**GDP**	**Product**
D	*1452200*	*steel*
D	*1452200*	*coal*
D	*1452200*	*chemicals*
D	*1452200*	*machinery*
D	*1452200*	*vehicles*
CH	*158500*	*machinery*
CH	*158500*	*chemicals*
CH	*158500*	*watches*
⋮	⋮	⋮

$\mathcal{F} = \{\textit{(Code,Product)} \to \textit{(Code, Product, GDP)},\ \textit{Code} \to \textit{GDP}\}$

Key: (Code, Product) □

2ND NORMAL FORM (2NF)

- In Example 7.20, the GDP information (e.g., (*D, 1452200*)) is stored redundantly.
- Problem: Code $\to$ GDP, but Code alone is not a key.

2NF forbids non-trivial FDs, where a non-key attribute A is functionally dependent on some $\bar{X}$ that is a proper subset of a key. Such FDs cause the above redundancy.

Definition 7.9
A relation schema $R = (\bar{V}, \mathcal{F})$ is in 2NF if and only if every non-key attribute A is fully *dependent on each candidate key:*

- *Let $\bar{K}$ be a candidate key of R, A an attribute that is not contained in any candidate key. Then, there is no $\bar{X} \subsetneq \bar{K}$ s.t. $\bar{X} \to A \in \mathcal{F}$.* □

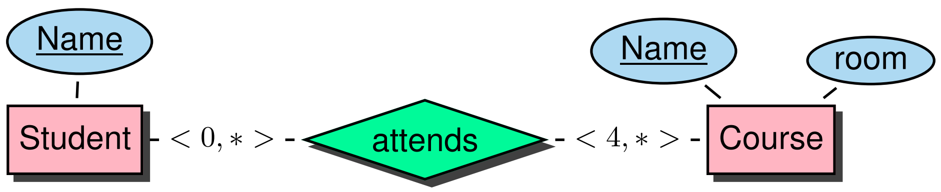
Example 7.21
Consider again Example 7.20: Split the Economy *relation into relations* Economy'(Code, GDP) *and* Products(Code, Product)*.* □

2ND NORMAL FORM (CONT'D)

The above example was motivated by normalizing a multivalued attribute.

The same situation can occur when mapping a multivalued relationship inaccurately:

- non-key attributes of one of the participating entity types is mixed with the relationship.



attends		
Student	Course	room
Alice	Databases	E105
Bob	Databases	E105
Alice	Telematics	E105
Carol	Telematics	E105
Bob	Programming	E203

(Student, Course) is (the only) candidate key.

$\mathcal{F} = \{$ *Course* $\to$ *Room*,
(Student, Course) $\to$ *Room* $\}$

- The table contains redundancies
- 2NF Decomposition: Separate the relationship from the entity.

2nd Normal Form (Cont'd)

Separate the relationship from the entity:

attends		
Student	Course	room
Alice	Databases	E105
Bob	Databases	E105
Alice	Telematics	E105
Carol	Telematics	E105
Bob	Programming	E203

split

attends'	
Student	Course
Alice	Databases
Bob	Databases
Alice	Telematics
Carol	Telematics
Bob	Programming

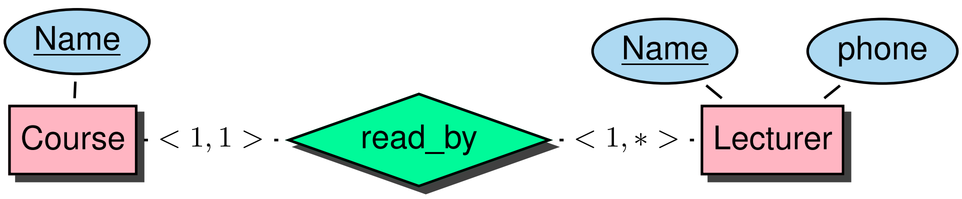
Course	
Name	room
Databases	E105
Telematics	E105
Programming	E203

Is that all?

No. The idea is clear, but the formulation is not yet perfectly accurate.

... 2NF covers only FDs from keys.

Consider the following situation when mapping a multivalued, $n:1$-relationship inaccurately:



read_by		
Course	Lecturer	phone
Telematics	Ho	14401
Mobile Comm	Ho	14401
Databases	WM	14412
SSD&XML	WM	14412

Course is (the only) candidate key.

$\mathcal{F} = \{$ *Course* $\rightarrow$ *Lecturer*
Course $\rightarrow$ *phone*
Lecturer $\rightarrow$ *phone* $\}$

- the table contains redundancies
- the table is in 2NF
- *Lecturer* $\rightarrow$ *phone* does not violate 2NF because Lecturer is not contained in any candidate key – this case is not covered by 2NF.

3RD NORMAL FORM (3NF)

Definition 7.10
A relation schema $R = (\bar{V}, \mathcal{F})$ is in 3NF if and only if for each non-key attribute A:

- *For each $\bar{X} \to A \in \mathcal{F}$ such that A is not contained in any candidate key, $\bar{X}$ contains a candidate key.* □

Now, *all* FDs for non key A must be “complete key $\to A$”

3NF Decomposition: Split again.

Separate the relationship from the entity:

read_by

Course	Lecturer	phone
Telematics	Ho	14401
Mobile Comm	Ho	14401
Databases	WM	14412
SSD&XML	WM	14412

split:

read_by’

Course	Lecturer
Mobile Comm	Ho
Telematics	Ho
Databases	WM
SSD&XML	WM

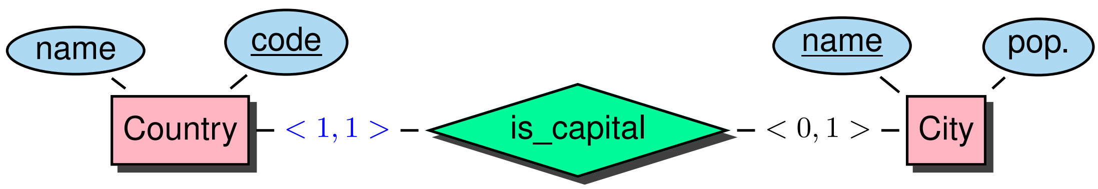
Lecturer

Lecturer	phone
Ho	14401
WM	14412

3NF-Decomposition is always lossless and dependency-preserving.

NORMAL FORMS

Compare: why can the relationship and the entity be combined in in the following case?



Boyce-Codd Normal Form (BCNF)

- In Example 7.19 (Languages), the name (e.g., *D, Germany*) is stored redundantly. (Note that *Name* is a key attribute there – thus 3NF is not applicable.)

BCNF extends 3NF for key attributes:

Definition 7.11
A relation schema $R = (\bar{V}, \mathcal{F})$ is in BCNF if and only if for each attribute A:

- *For each $\bar{X} \to A \in \mathcal{F}$ such that $A \notin \bar{X}$, $\bar{X}$ contains a key.* □

Example 7.22
Consider again Example 7.19: Name *depends on* Code*, but* Code *does not contain a key.*

Split the Languages *relation into relations* Country(<u>Code</u>,Name) *and* Languages'(<u>Code</u>,<u>Language</u>,Percent)*.*

In this case, the decomposition is lossless and dependency-preserving. □

BCNF (Cont'd)

- BCNF-Decomposition is always lossless, but not necessarily dependency-preserving.

Example 7.23
Consider again Example7.18:

$R = (\bar{V}, \mathcal{F})$*, where* $\bar{V} = \{$*City, Address, Zip*$\}$*, and* $\mathcal{F} = \{($*City,Address*$) \to$ *Zip*$,$ *Zip* $\to$ *City*$\}$.

R is in 3NF, but not in BCNF.

The decomposition R_1(<u>Address, Zip</u>) *and* R_2(City,<u>Zip</u>) *transforms it in a BCNF schema.*

It has been shown that this decomposition is lossless, but not dependency-preserving. □

Properties of BCNF and 3NF

Theorem 7.6
If a relation schema R has exactly one key, then R is in BCNF if and only if R is in 3NF.

Proof: *Obviously, BCNF implies 3NF. Assume R in 3NF and $\bar{K}$ its only key. Assume a FD $\bar{X} \to A \in \mathcal{F}$.*

We show that $\bar{X} \to A$ is trivial (i.e., $A \in \bar{X}$). Since R is in 3NF, it is sufficient to consider the case where A is a key attribute.

$(\bar{K} - A) \cup \bar{X}$ is a superkey (since $\bar{X} \to A$ and A is part of $\bar{K}$). Thus, there is a key $\bar{K}' \subseteq (\bar{K} - A) \cup \bar{X}$. Since there is only a single key, $\bar{K} = \bar{K}'$. Thus, since $A \in \bar{K}$, also $A \in K'$ – thus it must be in $\bar{X}$. □

Properties of BCNF and 3NF (Cont'd)

Lemma 7.4
A relation schema $R = (\bar{V}, \mathcal{F})$ is in BCNF if and only if for each non-trivial FD $\bar{X} \to A \in \mathcal{F}^+$, $\bar{X}$ is a superkey.

Proof:

- *"if" is obvious.*
- *It will be shown that if $\bar{X} \to A \in \mathcal{F}^+$ and $A \notin \bar{X}$, then $\bar{X} \to \bar{V} \in \mathcal{F}^+$.*

  Since $A \in \bar{X}^+ \setminus \bar{X}$, there is a non-trivial FD $\bar{Y} \to A \in \mathcal{F}$ that is used by the $\bar{X}^+$-algorithm for adding A to $\bar{X}^+$. For this, $\bar{Y} \subseteq \bar{X}^+$, i.e., $\bar{X} \to \bar{Y} \in \mathcal{F}^+$.

  Since R is in BCNF, $\bar{Y}$ is a superkey. Since $\bar{X} \to \bar{Y} \in \mathcal{F}^+$, $\bar{X}$ must already be a superkey – i.e., $\bar{X} \to \bar{V} \in \mathcal{F}^+$. □

Corollary 7.2
A relation schema $R = (\bar{V}, \mathcal{F})$ is in BCNF if and only if $R' = (\bar{V}, \mathcal{F}^+)$ is in BCNF. □

- Lemma 7.4 and Corollary 7.2 analogously hold for 3NF.

PRACTICAL ASPECTS

- BCNF can be tested in polynomial time.
  Sketch: Use the $\bar{X}^+$-algorithm for each FD $\bar{X} \to \bar{Y}$ to check if $\bar{X}$ is a superkey.
- Testing 3NF is NP-complete
  - polynomially check if BCNF – if “yes”, OK
  - if ”no”, the check whether A is a key attribute is exponential.
- Consider a set $\mathcal{F}$ of FDs over $\bar{V}$, and $\bar{X} \subseteq \bar{V}$.
  Then, for computing $\pi[\bar{X}](\mathcal{F})$, only algorithms are known that are (in the worst case) exponential in $|\bar{X}|$.
  Sketch: For every $\bar{Y} \subseteq \bar{X}$, compute $\bar{Y}^+$ and add $\bar{Y} \to (\bar{Y}^+ \cap \bar{X})$ to $\pi[\bar{X}](\mathcal{F})^+$
  (no way to compute $\pi[\bar{X}](\mathcal{F})$ without the closure).

PRACTICAL ASPECTS (CONT’D)

Lemma 7.5
For a relation schema $R = (\bar{V}, \mathcal{F})$ s.t. there is a FD $\bar{X} \to \bar{Y}$ where $\bar{X} \cap \bar{Y} = \emptyset$, the decomposition $\rho = (R \setminus \bar{Y}, \overline{XY})$ is lossless. □

Proof *Proof: Use Corollary 7.1 (Slide 7.1):*
$(R \setminus \bar{Y}) \cap \overline{XY} = \bar{X}$, $\overline{XY} \setminus (R \setminus \bar{Y}) = \bar{Y}$, *and thus* $\bar{X} \to \bar{Y}$. □

... this can now be used for an algorithm.

7.3.1 BCNF-Analysis: lossless, but not dependency-preserving

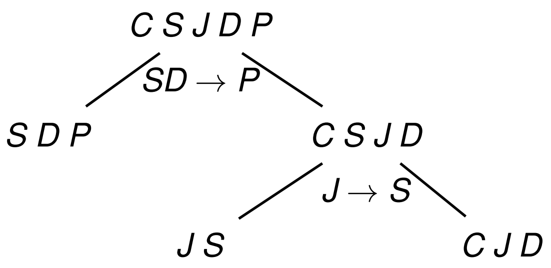
Input: a relation schema $R = (\bar{V}, \mathcal{F})$ that is not in BCNF.

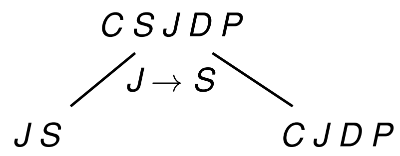
Consider a FD $\bar{X} \to A \in \mathcal{F}$ that violates the BCNF condition.

- Decomposition of $\bar{V}$: $\rho = (\overline{XA}, \bar{V} - A)$ $\quad$ (A has been stored redundantly)
- $R_1 = (\overline{XA}, \pi[\overline{XA}](\mathcal{F}))$
- $R_2 = (\bar{V} - A, \pi[\bar{V} - A](\mathcal{F}))$,
- check whether R_1 and R_2 satisfy the BCNF condition, apply algorithm recursively.

Example 7.24

Let $\bar{V}$ = {C,S,J,D,P}, $\mathcal{F}$ = {SD → P, J → S}.





SD → P is not preserved. □

7.3.2 3NF-Analysis: lossless and dependency-preserving

- Sketch: BCNF – and repair.

Consider a relation schema $R = (\bar{V}, \mathcal{F})$ such that

- $\mathcal{F}$ is minimal, and
- $\rho = (\bar{X}_1, \ldots, \bar{X}_k)$ is a decomposition of $\bar{V}$ such that all schemata $R_i = (\bar{X}_i, \pi[\bar{X}_i](\mathcal{F}))$ are in BCNF.
  (possibly not dependency-preserving)
- For each such FD $\bar{X} \to A$ that is not preserved, extend ρ with $\overline{XA}$; the corresponding schema is $(\overline{XA}, \pi[\overline{XA}](\mathcal{F}))$.
- The resulting decomposition is obviously lossless and additionally dependency-preserving. Each of the new schemata is in 3NF.

Proof Sketch: Since $\bar{X} \to A \in \mathcal{F}$ and $\mathcal{F}$ minimal, there is no $\bar{Y} \to A$ for any $\bar{Y} \subset \bar{X}$. Thus, $\bar{X}$ is a key for $\overline{XA}$ and all other FDs over $\overline{XA}$ are defined only over $\bar{X}$. Thus, they cannot violate the 3NF-condition (but the BCNF-condition).

Example 7.25

Consider again Example 7.24.

$\bar{V}$ = {C,S,J,D,P}, $\mathcal{F}$ = {SD $\rightarrow$ P, J $\rightarrow$ S}

- *The first decomposition is dependency-preserving.*
- *The second decomposition*

  C S J D P

  J $\rightarrow$ S

  J S   C J D P

  does not preserve SD $\rightarrow$ P.

  The 3NF-analysis algorithm adds S D P.

□

7.3.3 3NF-Synthesis: lossless and dependency-preserving

Input: relation schema $R = (\bar{V}, \mathcal{F})$ and $\mathcal{F}^{min}$.

1. Consider maximal sets of FDs from F^{min} with the same left hand side. Let $\{\bar{X} \rightarrow A_1,\ \bar{X} \rightarrow A_2,\ \ldots\}$ such a set.
   For every set, generate a schema with the format $\overline{XA_1A_2\ldots}$.
2. If none of the formats from (1) contains a key of R, take any key $\bar{K}$ of R and add a schema with format K.

- The 3NF-Synthesis-Algorithm is polynomial in time.
- the resulting ρ is not necessarily minimal:
  Consider $\bar{V} = \{AB\}$ with $\mathcal{F}^{min} = \{A \rightarrow B, B \rightarrow A\}$. Then, $\rho = (\underline{A}B, \underline{B}A)$.
- Recall that in contrast, it is NP-complete to check if a *given* schema is in 3NF.

Correctness

- Using $\mathcal{F}^{min}$, the generated schemata are in 3NF.
- ρ is dependency-preserving since for every $\bar{X} \to \bar{Y} \in \mathcal{F}^{min}$, a format is generated that contains $\overline{XY}$.
- ρ is lossless since ρ contains a key of the original schema. Using this tuple, in T^* (cf. Theorem 7.5) contains a row that consists of a_is:

  Consider the steps of the $\bar{X}^+$-algorithm that add – w.l.o.g. – the attributes $A_1, A_2, \ldots, A_k$ from $\bar{V} \setminus \bar{X}$ to $\bar{X}^+$. Show by induction that column of A_i in the row of $\bar{X}$ is set to a_i.
  - $i = 0$: nothing to show.
  - $i-1 \to i$: A_i is added to $\bar{X}^+$ due to a FD $\bar{Y} \to A_i$ where $\bar{Y} \subseteq \bar{X} \cup \{A_1, \ldots, A_{i-1}\}$. Furthermore, $\overline{YA_i} \subseteq \bar{X}'$ for some $\bar{X}' \in \rho$ (generated by step (1)) and the rows of $\bar{X}$ and $\bar{X}'$ coincide for $\bar{Y}$ (only as). Then, the chase copies the a_i from the row of $\bar{X}'$ to the row of $\bar{X}$.

7.4 Join Dependencies and Multivalued Dependencies

Example 7.26

Consider the following Non-1NF table:

cco		
Country	**Continents**	**Organizations**
D	*Europe*	*NATO, EU, UN*
TR	*Europe, Asia*	*NATO, UN*
R	*Europe, Asia*	*UN*
USA	*N.America*	*UN*

... expand the groups as before to 1NF ...

Join Dependencies and Multivalued Dependencies (Cont'd)

Example 7.26 (Continued)

the expanded table:

cco		
Country	**Continent**	**Organization**
D	Europe	NATO
D	Europe	EU
D	Europe	UN
TR	Europe	NATO
TR	Europe	UN
TR	Asia	NATO
TR	Asia	UN
R	Europe	UN
R	Asia	UN
USA	N.America	UN

There is some redundancy ...
called multivalued dependencies
cco satisfies

- *country* $\twoheadrightarrow$ *continent and*
- *country* $\twoheadrightarrow$ *organization.*

Join Dependencies and Multivalued Dependencies (Cont'd)

Example 7.26 (Continued)

cco		
Country	**Continent**	**Organization**
D	Europe	NATO
D	Europe	EU
D	Europe	UN
TR	Europe	NATO
TR	Europe	UN
TR	Asia	NATO
TR	Asia	UN
R	Europe	UN
R	Asia	UN
USA	N.America	UN

Actually, cco *is a join of*

encompasses	
Country	***Cont.***
D	Europe
TR	Europe
TR	Asia
R	Europe
R	Asia
USA	N.America

and

isMember	
Country	***Org.***
D	EU
D	NATO
D	UN
TR	UN
TR	NATO
R	UN
USA	UN

$$\begin{aligned} cco &= \pi[Country, Cont](cco) \bowtie \pi[Country, Org](cco) \\ &= encompasses \bowtie isMember \end{aligned}$$

□

JOIN DEPENDENCIES (CONT'D)

Consider a set $\bar{V}$ of attributes, a relation $r \in \text{Rel}(\bar{V})$, and a decomposition $\rho = \{\bar{X}_1, \ldots, \bar{X}_n\}$ of $\bar{V}$.

r satisfies the **join dependency (JD)** $\bowtie [\bar{X}_1, \ldots, \bar{X}_n]$ if and only if

$$r = \bowtie_{i=1}^{n} \pi[\bar{X}_i](r) \; .$$

In case that $n = 2$, the JD is also called a **multivalued dependency (MVD)**, written as

$$\bar{X}_1 \cap \bar{X}_2 \twoheadrightarrow \bar{X}_1 \setminus \bar{X}_2, \text{ or, symmetrically } \bar{X}_1 \cap \bar{X}_2 \twoheadrightarrow \bar{X}_2 \setminus \bar{X}_1 \; .$$

Note: $\bar{X}_1 = (\bar{X}_1 \cap \bar{X}_2) \cup (\bar{X}_1 \setminus \bar{X}_2)$, and $\bar{X}_2 = (\bar{X}_1 \cap \bar{X}_2) \cup (\bar{X}_2 \setminus \bar{X}_1)$.

7.4.1 4. Normal Form (4NF)

Goal: mutually independent facts should not be represented in a single relation.

Consider a relation schema $R = (\bar{V}, \mathcal{D})$ where $\mathcal{D}$ is a set of MVDs and FDs. Let $\mathcal{D}^+$ the closure of $\mathcal{D}$.

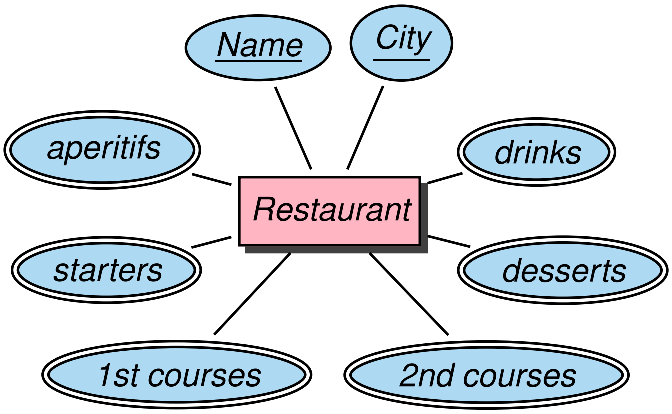
- for the closure $\mathcal{D}^+$ for MVDs see literature.
- FDs are special cases of MVDs.
- MVDs satisfy the following complement property:
  If $X \twoheadrightarrow Y \in \mathcal{D}^+$, then also $X \twoheadrightarrow (V \setminus (X \cup Y)) \in \mathcal{D}^+$.
- **trivial** MVDs are of the form $\bar{X} \twoheadrightarrow \bar{Y}$ for $\bar{Y} \subseteq \bar{X}$, and $\bar{X} \twoheadrightarrow V \setminus \bar{X}$.

Definition 7.12
A relation schema $R = (\bar{V}, \mathcal{D})$ is in 4NF if and only if for every non-trivial $\bar{X} \twoheadrightarrow Y \in \mathcal{D}^+$, $\bar{X}$ contains a key. □

Example 7.27
Consider again Example 7.26. It is not in 4NF.
Decomposition is lossless and dependency-preserving. □

Exercise 7.2

Experiment with join dependencies using the following ER diagram that describes restaurants that offer multiple choices of 2-course meals and accessoires (note that these attributes are multivalued):



7.5 Summary

- Analogous considerations for join dependencies lead to 5NF.
- 1NF $\Leftarrow$ (2NF) $\Leftarrow$ 3NF $\Leftarrow$ BCNF $\Leftarrow$ 4NF ($\Leftarrow$ 5NF)
  (other directions do not hold).
- 2NF is only of historical interest.
- In all cases there exists a lossless decomposition in 4NF (5NF).
- In the general case, all decompositions further than 3NF are not dependency-preserving.

7.6 Inclusion Dependencies

Consider sets $\bar{X}_1$ and $\bar{X}_2$ of attributes, and relations $r_1 \in \text{Rel}(\bar{X}_1)$ and $r_2 \in \text{Rel}(\bar{X}_2)$ with $\bar{Y} \subseteq \bar{X}_1 \cap \bar{X}_2$.

r_1, r_2 satisfy the **inclusion dependency (ID)** $R_1[\bar{Y}] \subseteq R_2[\bar{Y}]$ if and only if

$$\pi[\bar{Y}](r_1) \subseteq \pi[\bar{Y}](r_2)\ .$$

7.7 Schema Design

1. Generate an ER-model. This means a thorough discussion of the data engineers and the specialists of the application area.
2. Note that keys, functional dependencies, multivalued dependencies, and inclusion dependencies belong to this stage!

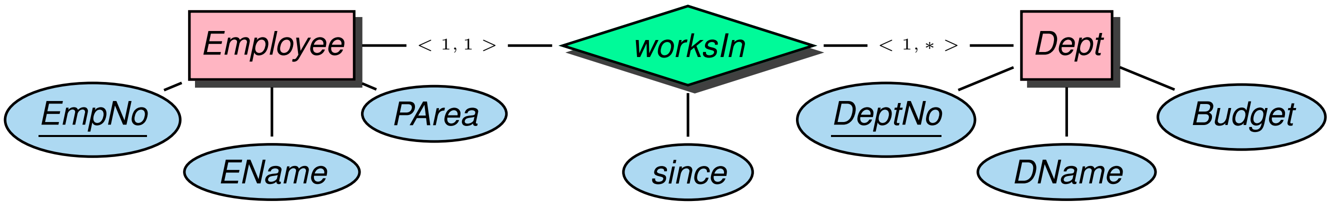
   Candidates can be found by data analysis, but the *semantic* aspect must be confirmed by the domain specialists.
3. Transformation to a relational schema
4. Normalization to 3NF
5. Manual decomposition to 4NF
6. enhanced ER design.

IMPORTANCE OF A CORRECT ER-DESIGN

Example 7.28

Employees are associated (uniquely) with departments. For every employee, the id, name, and the parking area must be stored. For each department, the name, the number, and the budget of the department are stored, together with the hiring date of each of the employees.

(A) An ER model:



(B) Dependency Analysis

The FD DeptNo → PArea is detected.

- *Inter-relational FDs are not allowed.*
- *the* universal relation *of a database (a broad join of all its relations along all FK-PK references) allows data analysis tools to detect such inter-relational FDs.*

⇒ *Re-Design* □