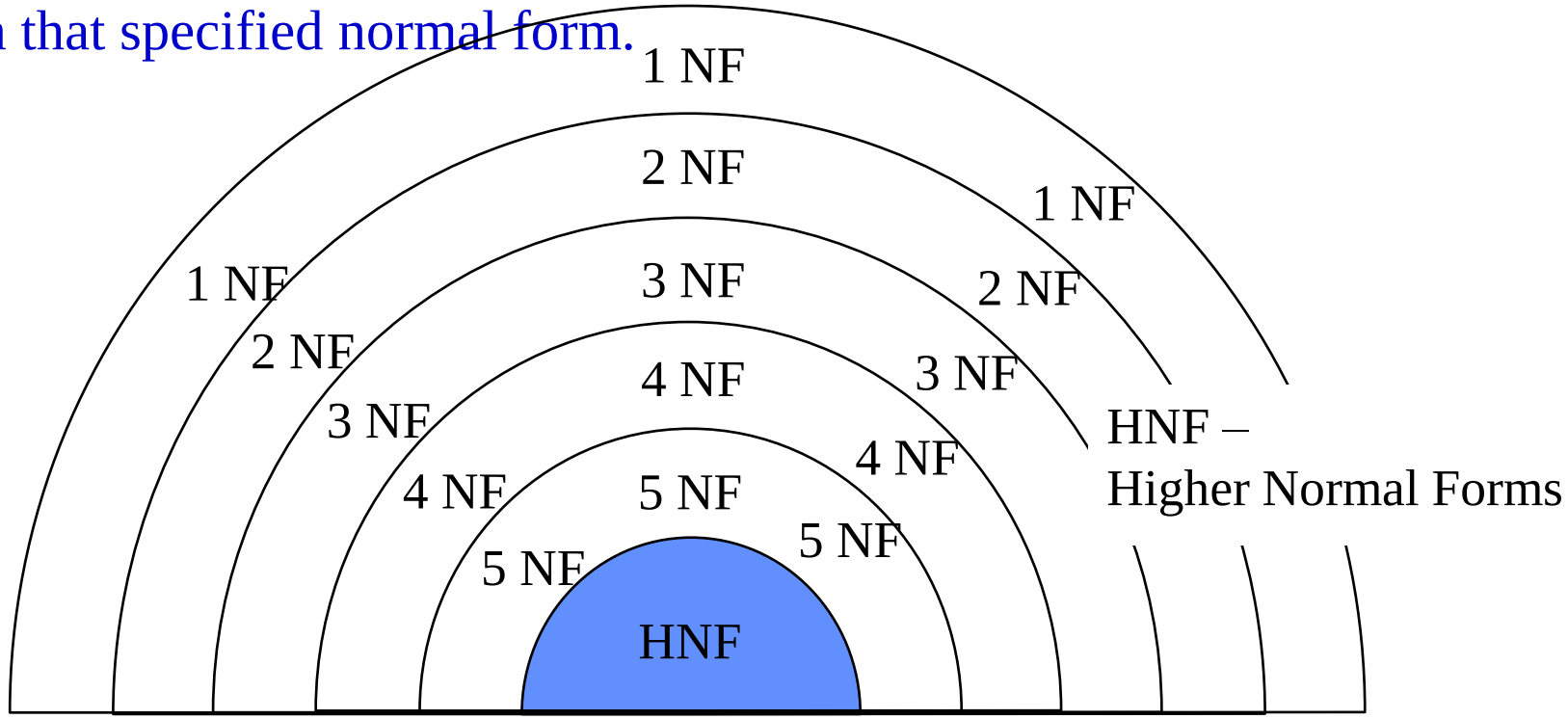
Database solutions

The process of normalization

Marzena Nowakowska
Faculty of Management and Computer Modelling
Kielce University of Technology
room: 3.21 C

The definition

A database normalisation is **the process** in which data are organized in tables in such a way that allows to reduce redundancy and improve data integrity. The **evaluation** of a table is done to determine whether it is in a specified normal form **and**, if necessary, the table is **transformed** (usually by decomposition) to tables in that specified normal form.



Functional dependency describes the relationship between attributes in a table. Atribute B is functionally dependent on A ($A \rightarrow B$) if each value of A is associated with exactly one value of B. A is a **determinant** of B.

First Normal Form – 1NF

A table is in **first normal form** if for each column (attribute) the intersection with each row (a table cell) contains only one value from the domain of this column.

For a table to be in 1NF it is necessary to identyfiy and to remove repeating groups within the table. The **repeating group** is an atribute (or attributes) that contain more than one value for a single key occurrence.

There are the following approaches of removing repeating groups.

A. Entering appropriate data in an additionally created row and duplicating the non-repeating data, where required (*redundancy appears*).
B. Moving data containing repeating groups along with a copy of the original key attribute into a separate table.

Patients			
Surname	**Forename**	**Birdth_date**	**Telephone**
Oblęgorski	Adam	1980-01-12	604 789 660; 41 334 56 89
Wójtowicz	Tamara	2000-05-16	897 678 564
Mackiewicz	Ewa	1980-01-12	786 888 575; 61 675 45 67

to 1NF – approach A

Repeating group

Redundancy

Patient_data			
Surname	**Forename**	**Birdth_date**	**Telephone**
Oblęgorski	Adam	1980-01-12	604 789 660
Oblęgorski	Adam	1980-01-12	41 334 56 89
Wójtowicz	Tamara	2000-05-16	897 678 564
Mackiewicz	Ewa	1980-01-12	786 888 575
Mackiewicz	Ewa	1980-01-12	61 675 45 67

to 1NF – approach B

Patients			
P_id	**Surname**	**Forename**	**Birdth_date**
1	Oblęgorski	Adam	1980-01-12
2	Wójtowicz	Tamara	2000-05-16
3	Mackiewicz	Ewa	1980-01-12

Patient_telephones	
P_id	**Telephone**
1	604 789 660
1	41 334 56 89
2	897 678 564
3	786 888 575
3	61 675 45 67

Transformation to 1 NF

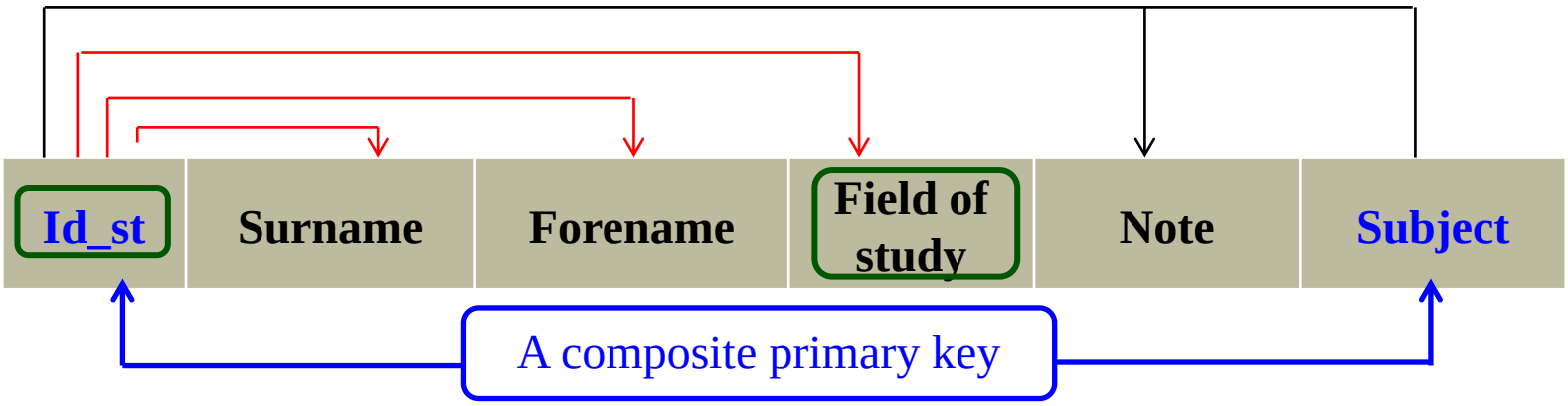
Second Normal Form – 2NF

A table is in **second normal form** if it is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key.

Attribute B is **fully functionally dependent** on A ={A1, A2} if B is functionally dependent on A but not on any proper subset of A ({A1}, {A2}). If there is a proper subset on which B depends, there is a partial dependence between attributes.

If the primary key of a table contains only a single attribute, the table is automatically in second normal form.

Id_st = identifier of a student



Dependencies in the table *Students_notes* under certain assumptions

In order to obtain second normal form, partial dependencies have to be removed – this is done via table decomposition.

Transformation to 2 NF

Students_notes

Id_st	Surname	Forename	Field	Subject	Note
1	Nowak	Anna	Logistics	Databases	5
2	Kowalski	Jan	Logistics	Databases	4,5
3	Kowalski	Andrzej	Economics	Databases	3,5
4	Podsiadło	Karol	Economics	Databases	4
1	Nowak	Anna	Logistics	Cargo technology	3,5
2	Kowalski	Jan	Logistics	Cargo technology	4
3	Kowalski	Andrzej	Economy	Microeconomy	5
4	Podsiadło	Karol	Economy	Microeconomy	3

to 2NF

Students

Id_st	Surname	Forename	Field
1	Nowak	Anna	Logistics
2	Kowalski	Jan	Logistics
3	Kowalski	Andrzej	Economy
4	Podsiadło	Karol	Economy

Students_notes

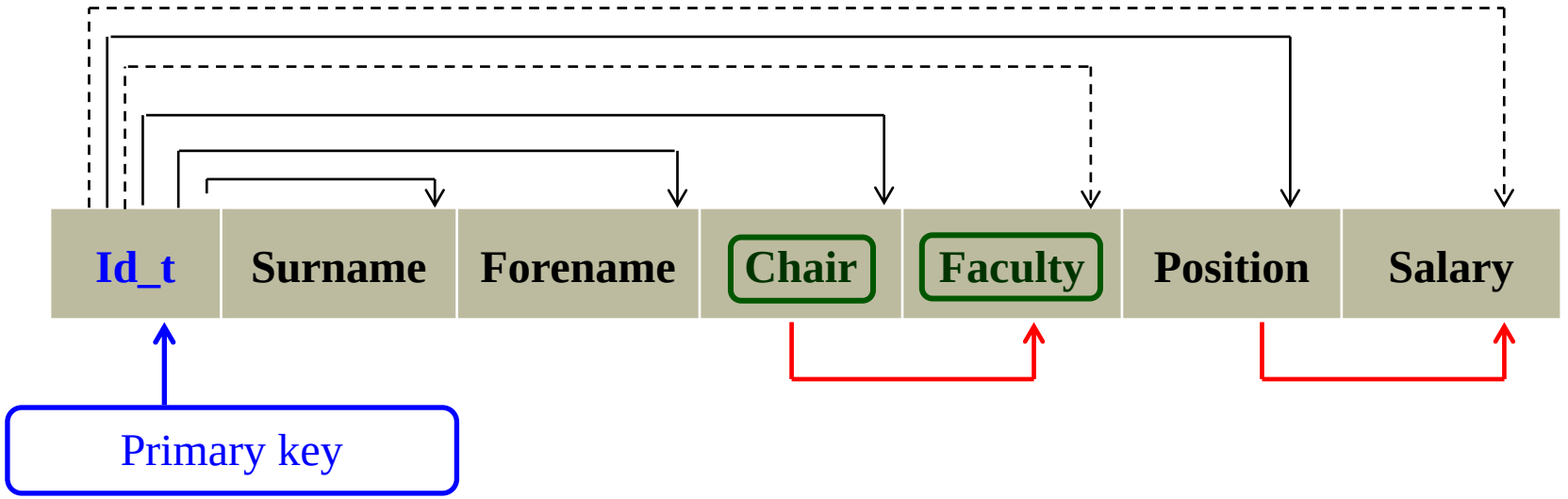
Id_st	Subject	Note
1	Databases	5
2	Databases	4,5
3	Databases	3,5
4	Databases	4
1	Cargo technology	3,5
2	Cargo technology	4
3	Microeconomy	5
4	Microeconomy	3

Third Normal Form - 3NF

A table is in **third normal form** if it is in second normal form and it has no transitive dependencies on the primary key.

Transitive dependency is a condition where A, B, C are attributes of a table, such that B is functionally dependent on A (A → B) and C is functionally dependent on B (B → C), and A is not functionally dependent on B or C. This is transitive dependency of C on A via B.

Id_t= identifier of a teacher



Dependencies in the table *Academic_teachers* under certain assumptions

Transformation to 3 NF

In order to obtain third normal form, transitive dependencies have to be removed – this is done via table decomposition.

Academic_teachers						
Id_t	**Surname**	**Forename**	**Chair**	**Faculty**	**Post**	**Salary**
1	Nowak	Anna	Mathematical methods	WZiMK	Professor	6500,00
2	Kowalski	Jan	Management	WZiMK	Adjunct	4050,00
3	Ferency	Andrzej	Metrology	WMiBM	Adjunct	4050,00
4	Podsiadło	Karol	Metrology	WMiBM	Asistant	3200,00

to 3NF

Academic_teachers				
Id_t	**Surname**	**Forename**	**Chair**	**Post**
1	Nowak	Anna	Mathematical methods	Professor
2	Kowalski	Jan	Management	Adjunct
3	Ferency	Andrzej	Metrology	Adjunct
4	Podsiadło	Karol	Metrology	Asistant

Chairs	
Chair	**Faculty**
Mathematical methods	WZiMK
Management	WZiMK
Metrology	WMiBM

Posts	
Post	**Salary**
Professor	6500,00
Adjunct	4050,00
Asistant	3200,00

Boyce-Codd Normal Form – BCNR (3.5NF)

A table is in **Boyce-Codd normal form** if and only if, with the exception of trivial functional dependency, every determinant is a candidate key. BCNR is stronger than 2NF and 3NF.

Trivial (obvious) functional dependency = dependency of an attribute on itself (a), **or** itself + something else (b).

Movies		
Release_year	**Movie_title**	**Director**
2008	The Dark Knight	Cristopher Nolan
2008	Iron Man	Jon Favreau
2008	Vicky Cristina Barcelona	Woody Allen
2010	127 Hours	Danny Boyle
2010	Toy Story 3	Lee Unkrich
2010	Unstoppable	Tony Scott

(a) {Release_year} → {Release_year}

(b) {Release_year, Movie_title} → {Release_year}

Informal definition of **Boyce-Codd normal form**

Every attribute in a table should depend on the key, the whole key, and nothing but the key.

Boyce-Codd Normal Form – BCNR example

Functional dependencies:

$\{Postal\_code\} \rightarrow \{Town\}$

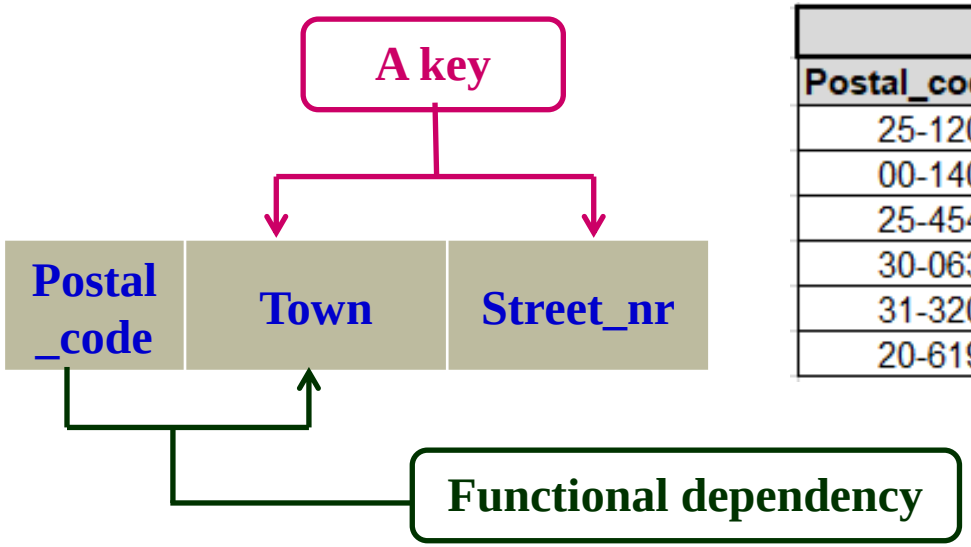
$\{Town, Street\_nr\} \rightarrow \{Postal\_code\}$

A composite key exists in the given below table (there is one candidate key):

$K = \{Town, Street\_nr\}$

In the first dependency *Postal_code* is a determinant, but not a (candidate) key

→ BCNF fails.



Adressess		
Postal_code	**Town**	**Street_nr**
25-120	Kielce	1000-lecia PP 7
00-140	Warszawa	Śliska 55/6
25-454	Kielce	Miła 6
30-063	Kraków	3-Maja 12/5
31-320	Kraków	Słowicza 4
20-619	Lublin	Nowomiejska 44

Fourth Normal Form 4NF

A table is in **fourth normal form** if it is in third normal form and if it contains no trivial multi-valued dependencies.

The **dependency** A → B is t**rivial** if B is the subset of A. Otherwise it is a **non-trivial dependency**. Examples:

- trivial dependency: {*Publication_year*, *Title*} → {*Title*}
- non-trivial dependency :{*Publication_year*, *Title*} → {*Price*}.

Multi-valued dependency is such a dependency between attributes A, B, and C in a table, in which for each values of A there is a set of values for B (A →→ B), and a set of values for C (A →→ C), however, B and C are independent of each other.

Id_s	Subject	Additional Activity

Id_s →→ *Subject* a student can participate in more than one subject
Id_s →→ *Additional Activity* a student can undertake more than one additional activity

Transformation to 4NF

Students_engagements		
Id_s	**Subject**	**Activity**
S1	Macroeconomy	AZS
S1	Microekonomy	Scientific association
S1	Microekonomy	AZS
S1	Macroeconomy	Scientific association
S2	Foreign markets	Student autonomy
S3	Macroeconomy	
S3	Accountancy	

Students_subjects	
Id_s	**Subject**
S1	Macroeconomy
S1	Microekonomy
S2	Foreign markets
S3	Macroeconomy
S3	Accountancy

Students_activities	
Id_s	**Activity**
S1	AZS
S1	Scientific association
S2	Scientific association

Fifth Normal Form

A table is in **fifth normal form** if it is in fourth normal form and if it has no join dependency, which means that there is no reunited decomposition of the table.

Fifth normal form refers to a table that can be decomposed but cannot be reconstructed losslessly (bezstratnie).

Lossless-join (also called non-loss or non-additive) dependency is a property of a decomposition, which ensures that no spurious (nieprawidziwy, błędny) rows are generated when tables are reunited through a natural join operation.

No join dependency

Shop_supply		
Warehouse	Shop	Good
H1	S1	T1
H1	S1	T2
H1	S2	T1
H1	S2	T3
H2	S1	T1
H2	S1	T3
H2	S3	T3

T1	
Warehouse	Shop
H1	S1
H1	S2
H2	S1
H2	S3

T2	
Warehouse	Good
H1	T1
H1	T2
H1	T3
H2	T1
H2	T3

T3	
Shop	Good
S1	T1
S1	T2
S1	T3
S2	T1
S2	T3

Possible basic decomposition of the table *Shop_supply*

T_12		
Warehouse	Shop	Good
H1	S1	T1
H1	S1	T2
H1	S1	T3
H1	S2	T1
H1	S2	T2
H1	S2	T3
H2	S1	T1
H2	S1	T3
H2	S3	T1
H2	S3	T3

T_13		
Warehouse	Shop	Good
H1	S1	T1
H1	S1	T2
H1	S1	T3
H1	S2	T1
H1	S2	T3
H2	S1	T1
H2	S1	T2
H2	S1	T3
H2	S3	T3

T_23		
Warehouse	Shop	Good
H1	S1	T1
H1	S1	T2
H1	S1	T3
H1	S2	T1
H1	S2	T3
H1	S3	T3
H2	S1	T1
H2	S1	T3
H2	S2	T1
H2	S2	T3
H2	S3	T3

Results of natural join of each pair of decomposed tables

T_123		
Warehouse	Shop	Good
H1	S1	T1
H1	S1	T2
H1	S1	T3
H1	S2	T1
H1	S2	T3
H2	S1	T1
H2	S1	T3
H2	S3	T3

Natural join of T_12 and T_3