Dependency Theory – Part 2

Relational Databases Basics

TRAINING CENTER

Just a quick reminder…

See the previous part for dependencies NF1-NFBC rely on.

Read and remember!

Multivalued dependency ($\{X\} \twoheadrightarrow \{Y\}$ or $\{X\} \twoheadrightarrow \{Y\}|\{Z\}$) – a dependency that forces a relation (with 3+ attributes at least), that has two tuples with the same value of one attribute, to have two more tuples with "cross-combination" of two other attribute values.

I know, it's hard ☹. But this **is** an extremely simplified representation.

Dependencies for the 4NF: multivalued dependency

A little bit simpler…

R

	X	Y	Z
t_1	X_1	Y_1	Z_1
t_2	X_1	Y_2	Z_2
t_3	X_1	Y_1	Z_2
t_4	X_1	Y_2	Z_1

These two tuples “existed initially”

Existence of these tuples is defined by multi-valued dependency

“Cross-combinations”

Y_1	Z_2
Y_2	Z_1

This is how the relation would look like (if not 1NF requirements)

X	Y	Z
X_1	Y_1, Y_2	Z_1, Z_2

Dependencies for the 4NF: multivalued dependency in real life

One applicant may apply to several faculties, i.e.:
{ua_applicant} ↠ {ua_faculty}

One faculty has several entrance exams, i.e.:
{ua_faculty} ↠ {ua_exam}

university_application

ua_applicant	ua_faculty	ua_exam
Ivanov I.I.	Math. faculty	Computer science
Ivanov I.I.	Math. faculty	Mathematics
Ivanov I.I.	Phys. faculty	Computer science
Petrov P.P.	Phys. faculty	Physics
Petrov P.P.	Math. faculty	Computer science
Petrov P.P.	Math. faculty	Mathematics
Sidorov S.S.	Phys. faculty	Computer science
Sidorov S.S.	Phys. faculty	Physics

For each applicant we have to add as many rows, as many exams there are on a faculty, this applicant applies to.

Dependencies for the 4NF

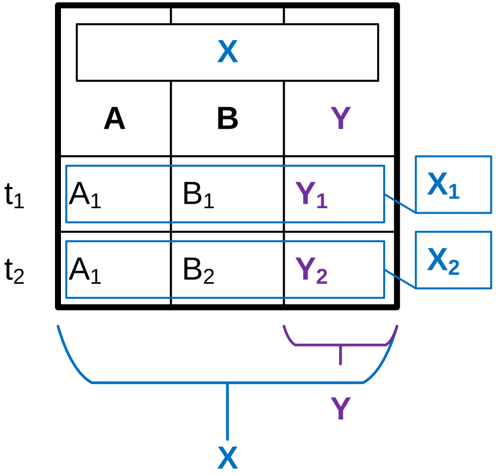
Read and remember!

Trivial multivalued dependency — multivalued dependency $\{X\} \twoheadrightarrow \{Y\}$, where Y is a subset of X, or the whole relation consists of X and Y only (i.e. $R = X \cup Y$).

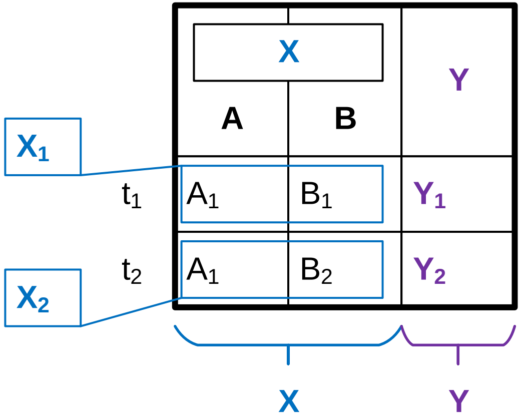
Nontrivial multivalued dependency — multivalued dependency $\{X\} \twoheadrightarrow \{Y\}$, where Y is NOT a subset of X, and the whole relation does NOT consist of X and Y only (i.e. $R \neq X \cup Y$).

Dependencies for the 4NF: trivial multivalued dependency

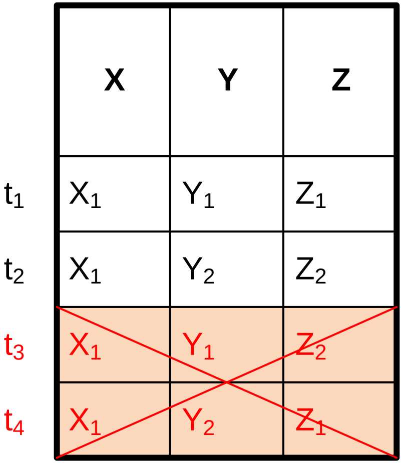
For nontrivial – skip two slides back.



Y is a subset of X



The whole relation consists of X and Y only



If $X \rightarrow Z$ or $Y \rightarrow Z$ dependency exists, t_3 and t_4 tuples can not exist (otherwise “one argument value – one function value” rule would be violated).

Dependencies for the 5NF

Read and remember!

Join dependency – a dependency $JD(R_1, R_2, ..., R_n)$ that forces a relation to comply with the rule: with any data inside the relation there should be possibility of lossless decomposition of R into R_1, R_2, ..., R_n projections.

Trivial join dependency – a dependency $JD(R_1, R_2, ..., R_n)$, in which at least one R_i component contains all relation attributes.

Nontrivial join dependency – a dependency $JD(R_1, R_2, ..., R_n)$, in which none of R_i components contain all relation attributes.

Dependencies for the 5NF: join dependency, example 1

elective

e_id	e_group	e_tutor	e_topic
157	1	33	Stop using ORMs!
248	8	24	MySQL vs …
411	1	78	Oracle 999 overview
489	7	33	Arduino in free space

Functional dependencies:
{e_id} → {e_group}
{e_id} → {e_tutor}
{e_id} → {e_topic}

Join dependency:
JD({e_id, e_group},
{e_id, e_tutor},
{e_id, e_topic})

$P_{1(e\_id,\ e\_group)}$

e_id	e_group
157	1
248	8
411	1
489	7

$P_{2(e\_id,\ e\_tutor)}$

e_id	e_tutor
157	33
248	24
411	78
489	33

$P_{3(e\_id,\ e\_topic)}$

e_id	e_topic
157	Stop using ORMs!
248	MySQL vs …
411	Oracle 999 overview
489	Arduino in free space

Dependencies for the 5NF: join dependency, example 2

workload

	w_tutor	w_subject	w_faculty
(1)	Ivanov I.I.	Mathematics	Exact sciences
(2)	Ivanov I.I.	Informatics	Natural science
(3)	Petrov P.P.	Mathematics	Natural science
(4)	Sidorov S.S.	Informatics	Cybernetics
(5)	Petrov P.P.	Physics	Exact sciences
(6)	Petrov P.P.	Mathematics	Exact sciences
(7)	Ivanov I.I.	Mathematics	Natural science

Functional dependencies: none.

Join dependency:
JD({w_tutor, w_subject},
{w_subject, w_faculty},
{w_tutor, w_faculty})

There is a rule: “If a tutor deliver some subject S, and some faculty F has that subject S, and this tutor works on this faculty, he has to deliver the subject S on this faculty F”.

a) There is “Mathematics” on “Exact sciences” faculty (1).
b) Petrov P.P. works on “Exact sciences” faculty (5).
c) Petrov P.P. delivers “Mathematics” (3).
FOLLOWS: Petrov P.P. has to deliver “Mathematics” on “Exact sciences” faculty (6).

Dependencies for the 5NF: join dependency, example 2

workload

	w_tutor	w_subject	w_faculty
(1)	Ivanov I.I.	Mathematics	Exact sciences
(2)	Ivanov I.I.	Informatics	Natural science
(3)	Petrov P.P.	Mathematics	Natural science
(4)	Sidorov S.S.	Informatics	Cybernetics
(5)	Petrov P.P.	Physics	Exact sciences
(6)	Petrov P.P.	Mathematics	Exact sciences
(7)	Ivanov I.I.	Mathematics	Natural science

Functional dependencies: none.

Join dependency:
JD({w_tutor, w_subject},
{w_subject, w_faculty},
{w_tutor, w_faculty})

There is a rule: "If a tutor deliver some subject S, and some faculty F has that subject S, and this tutor works on this faculty, he has to deliver the subject S on this faculty F".

a) There is "Mathematics" on "Natural science" faculty (3).
b) Ivanov I.I. works on "Natural science" faculty (2).
c) Ivanov I.I. delivers "Mathematics" (1).
FOLLOWS: Ivanov I.I. has to deliver "Mathematics" on "Natural science" faculty (7).

Dependencies for the 5NF: join dependency, example 2

workload

w_tutor	w_subject	w_faculty
Ivanov I.I.	Mathematics	Exact sciences
Ivanov I.I.	Informatics	Natural science
Petrov P.P.	Mathematics	Natural science
Sidorov S.S.	Informatics	Cybernetics
Petrov P.P.	Physics	Exact sciences
Petrov P.P.	Mathematics	Exact sciences
Ivanov I.I.	Mathematics	Natural science

$P_{1(w\_tutor,\ w\_subject)}$

w_tutor	w_subject
Ivanov I.I.	Mathematics
Ivanov I.I.	Informatics
Petrov P.P.	Mathematics
Sidorov S.S.	Informatics
Petrov P.P.	Physics

$P_{3(w\_tutor,\ w\_faculty)}$

w_tutor	w_faculty
Ivanov I.I.	Exact sciences
Ivanov I.I.	Natural science
Petrov P.P.	Natural science
Sidorov S.S.	Cybernetics
Petrov P.P.	Exact sciences

$P_{2(w\_subject,\ w\_faculty)}$

w_subject	w_faculty
Mathematics	Exact sciences
Informatics	Natural science
Mathematics	Natural science
Informatics	Cybernetics
Physics	Exact sciences

While non two of these projections are enough to restore the initial relation ("new" records appear, all three projections are enough.

Dependencies for the 5NF: join dependency, example 3

workload

w_tutor	w_subject	w_faculty
Ivanov I.I.	Mathematics	Exact sciences
Ivanov I.I.	Informatics	Natural science
Petrov P.P.	Mathematics	Natural science
Sidorov S.S.	Informatics	Cybernetics
Petrov P.P.	Physics	Exact sciences

Functional dependencies: none.

Join dependencies: none.

$P_{1(w\_tutor,\ w\_subject)}$

w_tutor	w_subject
Ivanov I.I.	Mathematics
Ivanov I.I.	Informatics
Petrov P.P.	Mathematics
Sidorov S.S.	Informatics
Petrov P.P.	Physics

$P_{2(w\_subject,\ w\_faculty)}$

w_subject	w_faculty
Mathematics	Exact sciences
Informatics	Natural science
Mathematics	Natural science
Informatics	Cybernetics
Physics	Exact sciences

$P_{3(w\_tutor,\ w\_faculty)}$

w_tutor	w_faculty
Ivanov I.I.	Exact sciences
Ivanov I.I.	Natural science
Petrov P.P.	Natural science
Sidorov S.S.	Cybernetics
Petrov P.P.	Exact sciences

Non of these projections are enough to restore the initial relation.

Dependencies for the DKNF

Read and remember!

Domain dependency – a dependency IN(A, S) that forces each and every value of A to be a member of S set.

Key dependency – a dependency KEY(K) that forces each and every K value to be unique.

Dependencies for the DKNF: domain dependency

event

e_date	e_dayofweek
2018-05-01	Tuesday
2018-05-01	Tuesday
2018-05-02	Wednesday
2018-05-02	Wednesday

Domain dependency holds

event

e_date	e_dayofweek
2018-05-01	Grapefruit
2018-05-01	Tuesday
2018-05-02	Wednesday
2018-05-02	Wednesday

Domain dependency is violated

Dependencies for the DKNF: key dependency

employee

e_passport	e_salary
AA123456	10 000
BB654321	12 000
CC112233	11 000
DD332211	10 500

Key dependency holds

employee

e_passport	e_salary
AA123456	10 000
AA123456	12 000
CC112233	11 000
DD332211	10 500

Key dependency is violated

Read and remember!

Generalized join dependency – a dependency USING(ACL): ¤ $\{X_1, X_2, ..., X_n\}$ that forces relation R to comply with the rule: join operation on $X_1, X_2, ..., X_n$ projections should produce a relation with the same header H as initial projection R had.

Horizontal decomposition – a decomposition of chronological relation into projections with explicit intervals ("from … to…") a with implicit intervals ("from … to NOW", "from NOW to …").

Vertical decomposition – a decomposition of chronological relation into such relations that each new relation is in 6NF.

Dependencies for the 6NF: generalized join dependency

education_path

<u>ep_student</u>	<u>ep_period</u>	ep_faculty	ep_group
13452	01.01.2018-31.05.2018	Chemistry	1
13452	01.03.2018-28.02.2018	Chemistry	5
13452	01.06.2018-01.12.2018	Physics	5

$P_{1(ep\_student,\ ep\_period,\ ep\_faculty)}$

<u>ep_student</u>	<u>ep_period</u>	ep_faculty
13452	01.01.2018-31.05.2018	Chemistry
13452	01.06.2018-01.12.2018	Physics

$P_{2(ep\_student,\ ep\_period,\ ep\_group)}$

<u>ep_student</u>	<u>ep_period</u>	ep_group
13452	01.01.2018-28.02.2018	1
13452	01.03.2018-01.12.2018	5

There is no possibility to decompose vertically P_1 and P_2 any further (at least one of the new projections will be exactly the same as the initial one), thus P_1 and P_2 has trivial generalized join dependencies (or, in other words: they have no non-trivial generalized join dependencies).

Dependencies for the 6NF: generalized join dependency

P_1

ep_student	ep_period	ep_faculty
13452	01.01.2018-31.05.2018	Chemistry
13452	01.06.2018-01.12.2018	Physics
13452	02.12.2018-NOW	Biology

But we can still decompose these projections horizontally. Here is the example for P_1.

P_{1_DURING}

ep_student	ep_period	ep_faculty
13452	01.01.2018-31.05.2018	Chemistry
13452	01.06.2018-01.12.2018	Physics

P_{1_NOW}

ep_student	ep_start	ep_faculty
13452	02.12.2018	Biology

This is the ultimate possible decomposition ever.

Dependencies for the 6NF: generalized join dependency

P_2

ep_student	ep_period	ep_group
13452	01.01.2018-28.02.2018	1
13452	01.03.2018-01.12.2018	5
13452	02.12.2018-NOW	10

But we can still decompose these projections horizontally. Here is the example for P_2.

P_{2_DURING}

ep_student	ep_period	ep_group
13452	01.01.2018-28.02.2018	1
13452	01.03.2018-01.12.2018	5

P_{2_NOW}

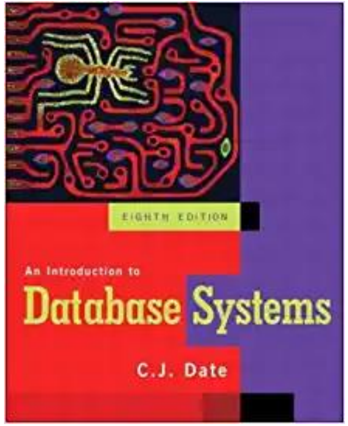
ep_student	ep_start	ep_group
13452	02.12.2018	10

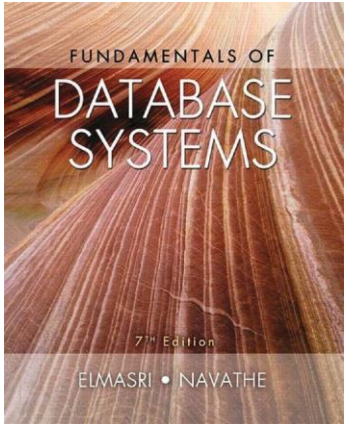
This is the ultimate possible decomposition ever.

That's all! ☺ But please remember…

Dependency theory is much wider, deeper and complex, than any quick video may ever cover.

Refer to these (or any other) books for more information.





Dependency Theory – Part 2

Relational Databases Basics

TRAINING CENTER