

Lista zadań nr 5 (normalizacja)

Boyc Danyś 1002

Posiadaję następującą bazę danych dla Wydziału Przemysłowej Zorganizowanej (WPZ).
Tę bazę danych opracowały już trzy najwybitniejsi naukowcy miast oraz najwybitniejsi znanowcy prawnicy. Wskazano też pewną ewentualną redundancję, że jeśli gang ugrasza jakiegoś człowieka to ugrasza go we wszystkich miastach, w których jest obecny, a jeśli jest obecny w jakimś mieście to ugrasza w nim wszystkie procedury, które ugrasza gdzieśkolwiek.

W tej chwili bieżąca sytuacja: **Małta (Mtaato, Gang, Proceder, Szef, Dochód, ROL)**. Wskazano też, że niechodzą niektóre funkcje to: Gang → Szef, Szef → Dochód, Proceder → ROL.

- (0,5 pkt.) Czy można tego, że baza (po Twoich modyfikacjach) jest w postaci BCNF? Zaproponuj taki rozkład bazy, aby postać BCNF, który jest najlepszy i zachowuje wszystkie zależności. Nie modyfikuj jakiegokolwiek tabeli, która już jest w postaci BCNF.
- (0,5 pkt.) Czy mimo tego, że baza (po Twoich modyfikacjach) jest w postaci BCNF dostatecznie w niej jeszcze jakieś redundancje? Jeśli jest jej trochę?
- (0,5 pkt.) Czy istnieje odniesienie rozkładu pozwalające się tej redundancji pozbyć? Podaj go. Dlaczego jest on odniesieniem?
- (0,5 pkt.) Napisałem sprytnie słaby rozkład zwracający nieprawidłowe odpowiedzi i widać wtedy gdy w tabeli Małta normalizacja jest zdefiniowana funkcją Proceder → ROL.

LINKI:

Prezentacja BCNF: <https://www.cs.pitt.edu/~cspon/papers/cic/bcnf/bcnf.pdf>

Zależność: https://en.wikipedia.org/wiki/Functional_dependency

BCNF (3Normal): https://en.wikipedia.org/wiki/Boyce%E2%8093Codd_normal_form

3normal: https://en.wikipedia.org/wiki/Third_normal_form

2normal: https://en.wikipedia.org/wiki/Second_normal_form

1normal: https://en.wikipedia.org/wiki/First_normal_form

i jeszcze ogólny opis: <https://not.mimuw.edu.pl/lecture.php?course=csad&week=146>

Co to BCNF?

Atrybut: wartość w tabeli niebędąca kluczem (lub jego częścią)

Funkcyjna zależność:

In relational database theory, a **functional dependency** is a **constraint** between two sets of attributes in a relation from a database. In other words, a functional dependency is a constraint between two attributes in a relation. Given a relation R and sets of attributes $X, Y \subseteq R$, it is said to **functionally determine** Y (written $X \twoheadrightarrow Y$) if and only if each X value in R is associated with precisely one Y value in R . R is then said to **satisfy** the functional dependency $X \twoheadrightarrow Y$. Equivalently, the **projection** $\Pi_{X,Y} R$ is a **function**, i.e. Y is a function of X .^[1] In simple words, if the values for the X attributes are known (say they are x), then the values for the Y attributes corresponding to x can be determined by looking them up in any table of R containing x . Customarily X is called the **determinant** set and Y the **dependent** set. A functional dependency $FD: X \twoheadrightarrow Y$ is called **trivial** if Y is a **subset** of X .

Przykład:

This example illustrates the concept of functional dependency. The situation modelled is that of college students visiting one or more lectures in each of which they are assigned a teaching assistant (TA). Let's further assume that every student is in some semester and is identified by a unique integer ID.

Student ID	Semester	Lecture	TA
1234	5	Numerical Methods	John
1221	4	Numerical Methods	Smith
1234	5	Visual Computing	Sam
1201	2	Numerical Methods	Peter
1201	2	Physics II	Simon

We notice that whenever two rows in this table feature the same StudentID, they also necessarily have the same Semester values. This basic fact can be expressed by a functional dependency:

• StudentID \twoheadrightarrow Semester

Note that if a row was added where the student had a different value of semester, then the functional dependency FD would no longer exist. This means that the FD is implied by the data (it is possible to have values that would invalidate the FD).

BCNF - jak jest spełniony to nie ma redundancji (nie przechowujemy tych samych danych kilka razy)

BCNF wymaga 3NF, 3NF wymaga 2NF, 2NF wymaga 1NF, więc po kolei:

I First Normal Form: tabela nie może zawierać relacji (tabel) jako wartości

First normal form (1NF) is a property of a **relation** in a **relational database**. A relation is in first normal form if and only if no attribute domain has relations as elements.^[1] Or more informally, that no table column can have tables as values. **Database normalization** is the process of representing a database in

II Second Normal Form: UID, lub więcej głównych UID gdzie każdy atrybut posiadałby niepowtarzalny UID (niezależny od całego klucza)

Second normal form (2NF) is a normal form used in **database normalization**. A relation is in the second normal form if it fulfills the following two requirements:
1. It is in first normal form.
2. It does not have any nonprime attribute that is functionally dependent on any proper subset of any candidate key of the relation (i.e. it lacks partial dependencies). A nonprime attribute of a relation is an attribute that is not a part of any candidate key of the relation.
For example, a relation (or table) is in 2NF if:
1. It is in 1NF and has a single attribute unique identifier (UID) (or which uses every non-key attribute is dependent on the whole UID), or
2. It is in 1NF and has a multi-attribute unique identifier, and every regular attribute (not part of the UID) is dependent on all attributes in the multi-attribute UID, not just any attribute (or part) of the UID.

III Third Normal Form: wszystkie atrybuty zależą tylko od klucza

"Jedyny (non-key [attribute]) must provide a fact about the key, the whole key, and nothing but the key"

Przykład

Tournament winners			
Tournament	Year	Winner	Winner's date of birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Because each row in the table needs to tell us who won a particular Tournament in a particular Year, the **composite key** {Tournament, Year} is a minimal set of attributes guaranteed to uniquely identify a row. That is, {Tournament, Year} is a candidate key for the table.

The breach of 3NF occurs because the non-prime attribute {Winner's date of birth} is transitively dependent on the candidate key {Tournament, Year} through the non-prime attribute Winner. The fact that Winner's date of birth is functionally dependent on Winner makes the table vulnerable to logical inconsistencies, as there is nothing to stop the same person from being shown with different dates of birth on different records.

In order to express the same facts without violating 3NF, it is necessary to split the table into two:

Tournament winners			
Tournament	Year	Winner	
Indiana Invitational	1998	Al Fredrickson	
Cleveland Open	1999	Bob Albertson	
Des Moines Masters	1999	Al Fredrickson	
Indiana Invitational	1999	Chip Masterson	

Winner's dates of birth		
Winner	Date of birth	
Chip Masterson	14 March 1977	
Al Fredrickson	21 July 1975	
Bob Albertson	28 September 1968	

I jak wszystko jest spełnione to jeszcze BCNF (3,5 normal form)

Definition [edit]

If a **relational schema** is in BCNF then all redundancy based on **functional dependency** has been removed,^[6] although other types of redundancy may still exist. A relational schema R is in Boyce–Codd normal form if and only if for every one of its **dependencies** $X \twoheadrightarrow Y$, at least one of the following conditions hold:^[6]

- $X \twoheadrightarrow Y$ is a **trivial** functional dependency ($Y \subseteq X$).
- X is a **superkey** for schema R .^[6]

Note that if a relational schema is in BCNF, then it is in 3NF.

Relacja jest w postaci normalnej Boyce'a-Codda wtedy i tylko wtedy, kiedy każdy jej atrybut zależy funkcjonalnie tylko od jej klucza głównego.

Definition (Postać normalna Boyce-Codda. BCNF)

Relacja R w zbiorze zależności funkcyjnych F jest w postaci normalnej Boyce-Codda, jeśli dla każdej niestrywialnej zależności $\alpha \twoheadrightarrow \beta$ ($\alpha \cap \beta = \emptyset$) zbiór α jest nadkluczem.

Example [edit]

The following relation does not satisfy 2NF because:

- There is a multi-attribute unique identifier: "Manufacturer" and "Model".
- {Manufacturer country} is functionally dependent (predictable) on {Manufacturer}.
- {Manufacturer country} is not part of a candidate key, so it is a non-prime attribute. (It is assumed that it is possible for two manufacturers in the same country to make a toothbrush with the same model name, so {Manufacturer country, Model} is not a candidate key even though in the current table the pair uniquely identify rows.)
- {Manufacturer} is a proper subset of the {Manufacturer, Model} candidate key.

In other words, since {Manufacturer country} is a non-prime attribute functionally dependent on a proper subset of a candidate key, the relation is in violation of 2NF.

Electric toothbrush models		
Manufacturer	Model	Manufacturer country
Forte	X-Prime	Italy
Forte	Ultraclean	Italy
Dent-o-Fresh	EZbrush	USA
Brushmaster	SuperBrush	USA
Kobayashi	ST-60	Japan
Hoch	Toothmaster	Germany
Hoch	X-Prime	Germany

To make the design conform to 2NF, it is necessary to have two relations. To create these relations:

- Remove the functionally dependent attributes in the partial dependencies of the first normal form relation. In this example, {Manufacturer country} is the functionally dependent attribute which will be removed.
- Place those partial dependency-dependent attributes (i.e. {Manufacturer country}) in a relation where their corresponding determinant attributes are a candidate key (i.e. {Manufacturer}).

As seen below, {Manufacturer country} is removed from the original table:

Electric toothbrush models

Manufacturer	Model
Forte	X-Prime
Forte	Ultraclean
Dent-o-Fresh	EZbrush
Brushmaster	SuperBrush
Kobayashi	ST-60
Hoch	Toothmaster
Hoch	X-Prime

As seen below, the partial dependency is put into a new relation where the dependency can exist without being a partial dependency:

Electric toothbrush manufacturers

Manufacturer	Manufacturer country
Forte	Italy
Dent-o-Fresh	USA
Brushmaster	USA
Kobayashi	Japan
Hoch	Germany

An example of a 3NF table that does not meet BCNF is:

Today's court bookings			
Court	Start time	End time	Rate type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

The table does not adhere to BCNF. This is because of the dependency Rate type \rightarrow Court in which the determining attribute Rate type – on which Court depends – (1) is neither a candidate key nor a *superset* of a candidate key and (2) Court is no *subset* of Rate type.

Dependency Rate type \rightarrow Court is respected, since a Rate type should only ever apply to a single Court.

The design can be amended so that it meets BCNF:

Rate types			Today's bookings			
Rate type	Court	Member flag	Court	Start time	End time	Member flag
SAVER	1	Yes	1	09:30	10:30	Yes
STANDARD	1	No	1	11:00	12:00	Yes
PREMIUM-A	2	Yes	1	14:00	15:30	No
PREMIUM-B	2	No	2	10:00	11:30	No
			2	11:30	13:30	No
			2	15:00	16:30	Yes

The candidate keys for the Rate types table are (Rate type) and (Court, Member flag); the candidate keys for the Today's bookings table are (Court, Start time) and (Court, End time). Both tables are in BCNF. When (Rate type) is a key in the Rate types table, having one Rate type associated with two different Courts is impossible, so by using (Rate type) as a key in the Rate types table, the anomaly affecting the original table has been eliminated.

zod.1/rozklad

Wypisz tabelę dekompozycji relacji Mafia (Miasto, Gang, Proceder, Szeft, Dochód, Złoty). Wskazanie, że nie zachodzi zależność funkcyjna: Gang \rightarrow Szeft, Szeft \rightarrow Dochód, Proceder \rightarrow Złoty.

1. (0.5 pkt.) Podaj dekompozycję postaci BCNF. Czy powyższa baza jest w tej postaci? Dlaczego? Zaproponuj taki rozkład bazy, który jest odnieśniony i zachowuje wszystkie zależności. Nie rozkładaj pojedynczych tabeli, które już same są postacią BCNF.

Mafia (Miasto, Gang, Proceder, Szeft, Dochód, Roi)

Mafia (Miasto, Gang, Proceder)

Złoty (Proceder, Roi)

Szeftowie (Gang, Szeft)

Zarobki (Szeft, Dochód)

- Rozkładem relacji R nazywamy zbiór relacji $\{R_1, \dots, R_k\}$ taki, że $R = R_1 \cup \dots \cup R_k$.
- Oznaczmy przez F^+ domknięcie zbioru zależności F (wszystkie zależności, które muszą zachodzić gdy zależności z F zachodzą).
- Dla F — zbioru zależności R , rzeknemy F na R , jest $F_1 := \{ \alpha \rightarrow \beta \in F^+ \mid \alpha, \beta \in R \}$.
- Dla r — stanu relacji R , stanem R jest $r_1 = \pi_R(r)$.
- Złączenie naturalne jest operacją przeciwną do rozkładu.
- Rozkład R na R_1, \dots, R_k jest odnaleziony, jeśli dla każdego poprawnego stanu r (spełniającego zależności F) zachodzi:

$$r = r_1 \bowtie r_2 \bowtie \dots \bowtie r_k$$
- Rozkład R na R_1, \dots, R_k zachowuje zależności, jeśli:

$$F^+ = (F_1 \cup F_2 \cup \dots \cup F_k)^+$$

zod.2/

2. (0.5 pkt.) Czy mimo tego, że baza (po Twoich modyfikacjach) jest w postaci BCNF dostrzegasz w niej jeszcze jakąś oczywistą redundancję? Jaka jest jej przyczyna?

Tak – mafia wykonuje wszystkie procedury w każdym miesiącu

zod.3/

3. (0.5 pkt.) Czy istnieje odnaleziony rozkład pozwalający się tej redundancji pozbyć? Podaj go. Dlaczego jest on odnaleziony?

Mafia (Miasto, Gang, Proceder)



Teren (Miasto, Gang), Branzia (Gang, Proceder)

zod.4/

4. (0.5 pkt.) Napisz zapytanie algebry relacji zwracające niepustą odpowiedź wtedy i tylko wtedy gdy w relacji Mafia naruszona jest zależność funkcyjna Proceder \rightarrow Roi.

Będzie naruszona gdy dla tego samego procederu będą dwa różne roi

o Mafia1.proceder = Mafia2.proceder and Mafia1.roi != Mafia2.roi (Mafia1 x Mafia2)

