

# Notatki powtórkowe z baz danych

AKA „pierwsza solówka na grzejniku”

Mikołaj Pietrek

## 1 Podstawy SQL

### 1.1 Schematy

#### 1.1.1 Ramowy szablon każdego zapytania

```
SELECT kolumna
FROM tabela
WHERE warunek
```

#### 1.1.2 Bardziej zaawansowany przykład (łączenie tabel, łączenie warunków)

```
SELECT t1.k1, t1.k2
FROM ( t1 INNER JOIN t2 ON t1.k3 = t2.k3 )
WHERE ( t1.k4 > 0 ) AND ( t1.k5 IS NULL )
```

#### 1.1.3 Proste (niekoniecznie sensowne) podzapytanie

```
SELECT podzap.k2
FROM
  ( SELECT * FROM ( t1 INNER JOIN t2 ON t1.k1 = t2.k1 ) ) AS podzap1
  INNER JOIN
  ( SELECT * FROM ( t3 INNER JOIN t4 ON t3.k1 = t4.k1 ) ) AS podzap2
ON
  podzap1.k1 = podzap2.k1
```

### 1.2 Operatory

#### 1.2.1 Porównawcze

=	równe
<>	różne
!=	różne
>	większe
<	mniej
>=	większe lub równe
<=	mniej lub równe

### 1.2.2 Logiczne

A AND B	koniunkcja
A OR B	alternatywa
NOT A	zaprzeczenie

### 1.2.3 Typowe dla SQL

BETWEEN A AND B	pomiędzy dwiema wartościami
LIKE '%wzorzec%'	dopasowanie ciągu znaków do wzorca
A NOT IN B	poza danym zbiorem
A IS (NOT) NULL	jest równy/różny NULL

### 1.3 Łączenie tabel

A JOIN B (ON C)	iloczyn kartezjański
A INNER JOIN B (ON C)	łączenie naturalne
A LEFT JOIN B (ON C)	do lewej strony dostawiona prawa
A RIGHT JOIN B (ON C)	do prawej strony dostawiona lewa
A FULL JOIN B (ON C)	wszystkie rekordy (jak LEFT+RIGHT)

### 1.4 Funkcje agregujące

MIN(A)	wartość minimalna
MAX(A)	wartość maksymalna
AVG(A)	średnia
SUM(A)	suma
COUNT(A)	ilość

### 1.5 Operacje na zbiorach wynikowych

A UNION B	suma ( $\cup$ )
A INTERSECT B	iloczyn ( $\cap$ )
A EXCEPT B	różnica ( $\setminus$ )

### 1.6 Porządkowanie zbioru wynikowego (tylko na końcu)

... ORDER BY A (ASC)	sortowanie (domyślnie rosnąco)
... ORDER BY A DESC	sortowanie malejąco
... GROUP BY A	grupowanie
... HAVING C	filtr końcowy, po wykonaniu właściwego zapytania
... LIMIT n	pierwsze $n$ wyników

## 2 Algebra relacji

$\sigma(A)$	SELECT * FROM A	selekcja
$\sigma_C(A)$	SELECT * FROM A WHERE C	selekcja warunkowa
$\pi_K(A)$	SELECT K FROM A	projekcja
$\pi_K(\sigma_C(A))$	SELECT K FROM A WHERE C	projekcja selekcji warunkowej
$\rho_N(R)$	R AS N	zmiana nazwy
$\gamma_G(R)$	GROUP BY G	grupowanie
$\gamma_{MAX(k)}(R)$	SELECT MAX(K) FROM R	agregacja
$\delta(R)$	DISTINCT...	usuwanie powtórzeń
$\tau_K(R)$	R SORT BY K (ASC)	sortowanie rosnąco
$\tau_{K \text{ DESC}}(R)$	R SORT BY K DESC	sortowanie malejąco
$R \cup S$	R UNION S	suma zbiorów
$R \cap S$	R INTERSECT S	iloczyn zbiorów
$R \setminus S$	R EXCEPT S	różnica zbiorów
$R \times S$	R JOIN S	iloczyn kartezjański
$R \bowtie S$	R INNER JOIN S	naturalne złączenie
$R \bowtie_C S$	R JOIN S ON C	złączenie warunkowe
$R \bowtie_C S$	R JOIN S ON C	złączenie warunkowe

## 3 Normalizacja

### 3.1 Definicje zależności funkcyjnych

#### 3.1.1 Nieformalnie (intuicyjnie)

Zależność kolumny B od kolumny A oznacza, że mając rekord z A możemy jednoznacznie wybrać rekord z B.  $A \rightarrow B$ . Przykładowo, nazwisko może zależeć od numeru PESEL (do każdego numeru jest przypisane konkretne nazwisko). Natomiast klucz potencjalny to najmniejszy zestaw kolumn, który jednoznacznie zidentyfikuje każdy rekord tabeli.

1NF - wartość każdego rekordu jest niepodzielna i opisuje jeden obiekt.

2NF - jak 1NF oraz żadna kolumna nie zależy od klucza potencjalnego.

3NF - jak 1NF oraz żadna kolumna nie zależy od żadnej innej kolumny.

BCNF - jak 1NF oraz każda kolumna zależy od dowolnego klucza, albo jest podzbiorem klucza.

#### 3.1.2 Zależnościami, na podstawie „Podstawowego Kursu Systemów Baz Danych”

3NF - dla każdej nietrywialnej zależności, albo lewa strona zawiera klucz, albo prawa składa się z samych atrybutów podstawowych (należących do jakiegoś klucza).

BCNF - dla każdej nietrywialnej zależności lewa strona zawiera klucz.

#### 3.1.3 Formalnie (K.Kleczkowski)

1NF - Każdy atrybut jest atomowy.

2NF - Relacja jest w 1NF i żaden atrybut niekluczowy nie jest zależny od właściwego podzbioru dowolnego klucza kandydującego.

3NF - Relacja jest w 2NF i żaden atrybut niekluczowy nie jest zależny przechodnio od dowolnego klucza kandydującego.

BCNF - Relacja jest w 3NF i każda zależność funkcyjna  $X \rightarrow Y$  jest trywialna (tzn. dla  $Y \subseteq X$  jest  $X \rightarrow Y$ ), lub  $X$  jest nadkluczem (nadzbiorem dowolnego klucza kandydującego) relacji  $R$ .

## 3.2 Aksjomaty

### 3.2.1 Podstawowe

- zwrotność:  $(Y \subseteq X) \Rightarrow (X \rightarrow Y)$
- poszerzenie:  $(X \rightarrow Y) \Rightarrow (WX \rightarrow WY)$
- przechodniość:  $(X \rightarrow Y) \wedge (Y \rightarrow Z) \Rightarrow (X \rightarrow Z)$

### 3.2.2 Wynikające z podstawowych

- dekompozycja:  $(X \rightarrow YZ) \Rightarrow (X \rightarrow Y) \wedge (X \rightarrow Z)$
- kompozycja:  $(X \rightarrow Y) \wedge (W \rightarrow Z) \Rightarrow (XW \rightarrow YZ)$
- suma:  $(X \rightarrow Y) \wedge (X \rightarrow Z) \Rightarrow (X \rightarrow YZ)$
- pseudopzechodniość:  $(X \rightarrow Y) \wedge (YW \rightarrow Z) \Rightarrow (XW \rightarrow Z)$
- samozależność:  $(X \rightarrow X)$

## 3.3 Algorytm znajdowania domknięcia (K.Kleckowski)

1. Zapisujemy dotychczasowy zbiór atrybutów, który rozpinamy.
2. Sprawdzamy, czy jakiś podzbiór tych atrybutów pozwala obliczyć dodatkowe atrybuty na podstawie jakiejś zależności funkcyjnej. Jeśli znaleźliśmy taką zależność funkcyjną, która oblicza nam nowe atrybuty, zapisujemy prawą stronę tej zależności do naszego domknięcia.
3. Powtarzamy krok 2. (łącznie z dopisanymi wcześniej atrybutami), aż żadna zależność nie da nam nowych atrybutów do domknięcia.
4. Żądanym zbiorem jest domknięcie.

## 3.4 Minimalne klucze relacji (na podstawie rozwiązań K.Kleckowskiego)

1. Znajdujemy minimalne pokrycie poprzez usunięcie zależności, które można uzyskać z pozostałych.
2. Wybieramy takie atrybuty, które w minimalnym pokryciu nie występują po prawej stronie żadnej zależności funkcyjnej. Następnie rozpinamy je i dodajemy najmniejsze podzbiory dopełnienia tego domknięcia. Uzyskane klucze są minimalne.