

# Bazy danych i systemy zarządzania

---

## Wykład V

# Operacje algebraiczne w relacyjnych bazach danych Część I

---

# Definiowanie tablicy relacyjnej bazy danych

---

## Definiowanie tabel

Aby zdefiniować tablicę reprezentującą relację w bazie danych należy określić następujące elementy:

1. **Schemat tabeli** postaci  $R(A_1, A_2, \dots, A_n)$ , tzn.:
  - zdefiniować sekwencję atrybutów  $A_1, A_2, \dots, A_n$ ,
  - zdefiniować ich dziedziny  $D_1, D_2, \dots, D_n$  (określając typ, rozmiar, więzy spójności, etc. lub poprzez jawne (ekstensjonalne) zdefiniowania elementów każdego zbioru  $D_i$ ,  $i = 1, 2, \dots, n$ ,
  - zdefiniować klucz lub indeks jednoznaczny (opcjonalnie).
2. **Krotki relacji (rekordy tabeli)**, co można uczynić:
  - *ekstensjonalnie*, np. wpisując lub wczytując ze zbioru,
  - *intensjonalnie*, definiując warunek selekcji oraz zbiór (generator) potencjalnych krotek (uniwersum), np. wczytać dane z odpowiedniego zbioru z wykorzystaniem filtru.

## Predykat relacji

Predykatem relacji  $R \subseteq D_1 \times D_2 \times \dots \times D_n$  nazywamy funkcję postaci:

$$R \longrightarrow \{TRUE, FALSE\}$$

przy czym:

–  $\|R(d_1, d_2, \dots, d_n)\| = TRUE$  wtw. gdy  $(d_1, d_2, \dots, d_n) \in R$ , oraz

–  $\|R(d_1, d_2, \dots, d_n)\| = FALSE$  wtw. gdy  $(d_1, d_2, \dots, d_n) \notin R$ .

$\|R(d_1, d_2, \dots, d_n)\|$  oznacza *wartość logiczną* formuły atomowej  $R(d_1, d_2, \dots, d_n)$ .

## Suma relacji zgodnych

Operację sumy (unii) relacji można przenieść bezpośrednio z algebry zbiorów; relacje są bowiem zbiorami. Jednak ze względu na specyficzną strukturę elementów tych zbiorów oraz wynikające z nich możliwości reprezentacji w tabelach, sensownie jest zdefiniować sumę dla *relacji zgodnych*, tzn. relacji o identycznym schemacie.

Niech  $R(A_1, A_2, \dots, A_n)$  oraz  $S(A_1, A_2, \dots, A_n)$  będą dowolnymi relacjami zgodnymi, określonymi w uniwersum  $U = D_1 \times D_2 \times \dots \times D_n$ ,  $R \subseteq U$ ,  $S \subseteq U$ . Suma relacji zgodnych definiowana jest jako suma zbiorów  $R \cup S$ :

$$R \cup S = \{(d_1, d_2, \dots, d_n) \in U : (d_1, d_2, \dots, d_n) \in R \vee (d_1, d_2, \dots, d_n) \in S\}.$$

Sumę relacji  $R \cup S$  można też zadać intensjonalnie w  $U$  poprzez zdefiniowanie predykatu relacji jak następuje:

$$\|R \cup S(d_1, d_2, \dots, d_n)\| = \|R(d_1, d_2, \dots, d_n)\| \vee \|S(d_1, d_2, \dots, d_n)\|.$$

### Przykład

A	B	C	∪	A	B	C	=	A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>		a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	a <sub>4</sub>	b <sub>4</sub>	c <sub>4</sub>	a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>		
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>	a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>	a <sub>4</sub>	b <sub>4</sub>	c <sub>4</sub>		
a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>	a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>	a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>		
			a <sub>7</sub>	b <sub>7</sub>	c <sub>7</sub>	a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>		
						a <sub>7</sub>	b <sub>7</sub>	c <sub>7</sub>		

Zgodnie z prawami algebry zbiorów, duplikaty rekordów nie są uwzględniane (standardowo). Dla poprawnej realizacji operacji liczba i typy kolumn (atrybutów) powinny być identyczne. Liczba elementów relacji wynikowej  $T = R \cup S$  spełnia warunek  $\text{card}(T) \leq \text{card}(R) + \text{card}(S)$ .

## Różnica relacji zgodnych

Operację różnicy (odejmowania) relacji można przenieść bezpośrednio z algebry zbiorów (relacje są zbiorami). Jednak ze względu na specyficzną strukturę elementów tych zbiorów oraz wynikające z nich możliwości reprezentacji w tabelach, sensownie jest zdefiniować różnicę dla *relacji zgodnych*, tzn. relacji o identycznym schemacie.

Niech  $R(A_1, A_2, \dots, A_n)$  oraz  $S(A_1, A_2, \dots, A_n)$  będą dowolnymi relacjami zgodnymi, określonymi w uniwersum  $U = D_1 \times D_2 \times \dots \times D_n$ ,  $R \subseteq U$ ,  $S \subseteq U$ . Różnica relacji zgodnych definiowana jest jako różnica zbiorów  $R \setminus S$ :

$$R \setminus S = \{(d_1, d_2, \dots, d_n) \in U : (d_1, d_2, \dots, d_n) \in R \wedge (d_1, d_2, \dots, d_n) \notin S\}.$$

Różnicę relacji  $R \setminus S$  można też zadać intensjonalnie w  $U$  poprzez zdefiniowanie predykatu relacji jak następuje:

$$\|R \setminus S(d_1, d_2, \dots, d_n)\| = \|R(d_1, d_2, \dots, d_n)\| \wedge \neg \|S(d_1, d_2, \dots, d_n)\|.$$

### Przykład

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>
a <sub>4</sub>	b <sub>4</sub>	c <sub>4</sub>
a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>
a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>
a <sub>7</sub>	b <sub>7</sub>	c <sub>7</sub>

\

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>
a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>
a <sub>8</sub>	b <sub>8</sub>	c <sub>8</sub>
a <sub>9</sub>	b <sub>9</sub>	c <sub>9</sub>

=

A	B	C
a <sub>4</sub>	b <sub>4</sub>	c <sub>4</sub>
a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>
a <sub>7</sub>	b <sub>7</sub>	c <sub>7</sub>

Dla poprawnej realizacji operacji odejmowania liczba i typy kolumn (atrybutów) powinny być identyczne. Wynikiem odejmowania  $R \setminus S$  jest relacja pusta wtw. gdy  $R \subseteq S$ . Liczba elementów relacji wynikowej  $T = R \setminus S$  spełnia warunek  $\text{card}(T) \leq \text{card}(R)$ .

## Iloczyn algebraiczny (przecięcie) relacji zgodnych

Operację iloczynu algebraicznego (zwykłego) (przecięcia) relacji można przenieść bezpośrednio z algebry zbiorów (relacje są zbiorami). Jednak ze względu na specyficzną strukturę elementów tych zbiorów oraz wynikające z nich możliwości reprezentacji w tabelach sensowniej jest zdefiniować iloczyn dla *relacji zgodnych*, tzn. relacji o identycznym schemacie.

Niech  $R(A_1, A_2, \dots, A_n)$  oraz  $S(A_1, A_2, \dots, A_n)$  będą dowolnymi relacjami zgodnymi, określonymi w uniwersum  $U = D_1 \times D_2 \times \dots \times D_n$ ,  $R \subseteq U$ ,  $S \subseteq U$ . Iloczyn relacji definiowany jest jako iloczyn zbiorów  $R \cap S$ , tzn.:

$$R \cap S = \{(d_1, d_2, \dots, d_n) \in U : (d_1, d_2, \dots, d_n) \in R \wedge (d_1, d_2, \dots, d_n) \in S\}.$$

Iloczyn relacji  $R \cap S$  można też zadać intensjonalnie w  $U$  poprzez zdefiniowanie predykatu relacji jak następuje:

$$\|R(\cap S d_1, d_2, \dots, d_n)\| = \|R(d_1, d_2, \dots, d_n)\| \wedge \|S(d_1, d_2, \dots, d_n)\|.$$

### Przykład

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>
a <sub>4</sub>	b <sub>4</sub>	c <sub>4</sub>
a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>
a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>
a <sub>7</sub>	b <sub>7</sub>	c <sub>7</sub>

 $\cap$ 

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>
a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>
a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>
a <sub>8</sub>	b <sub>8</sub>	c <sub>8</sub>
a <sub>9</sub>	b <sub>9</sub>	c <sub>9</sub>

 $=$ 

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>
a <sub>5</sub>	b <sub>5</sub>	c <sub>5</sub>
a <sub>6</sub>	b <sub>6</sub>	c <sub>6</sub>

Dla poprawnej realizacji operacji iloczynu liczba i typy kolumn (atrybutów) powinny być identyczne. Wynikiem operacji iloczynu jest relacja pusta wtw. gdy żaden rekord nie jest wspólny dla obu relacji. Liczba elementów relacji wynikowej  $T = R \cap S$  spełnia warunek  $card(T) \leq \min(card(R), card(S))$ . Iloczyn relacji można wyznaczyć korzystając z operacji różnicy wg wzoru:  $R \cap S = R \setminus (R \setminus S)$ .

## Dopełnienie relacji

Operację dopełnienia (uzupełnienia) relacji można przenieść bezpośrednio z algebry zbiorów (relacje są zbiorami), o ile określone jest uniwersum  $U$ , takie że  $R \subseteq U$ . Dopełnienie relacji  $R$  jest relacją o identycznym schemacie.

Niech  $R(A_1, A_2, \dots, A_n)$  będzie relacją określoną w uniwersum  $U = D_1 \times D_2 \times \dots \times D_n$ ,  $R \subseteq U$ . Dopełnienie relacji  $R$  definiowane jest jako relacja  $\overline{R}$  będąca uzupełnieniem  $R$  do uniwersum  $U$ , tzn.:

$$R = \{(d_1, d_2, \dots, d_n) \in U : (d_1, d_2, \dots, d_n) \notin R\}.$$

Dopełnienie relacji  $R$  można też zadać intensjonalnie w  $U$  poprzez zdefiniowanie predykatu relacji jak następuje:

$$\|\overline{R}(d_1, d_2, \dots, d_n)\| = \neg \|R(d_1, d_2, \dots, d_n)\|.$$

W praktyce, jeżeli dziedziny atrybutów  $A_1, A_2, \dots, A_n$  nie są zadane jawnie, przyjmuje się, że są one wyznaczone jako dziedziny relacji ze względu na odpowiednie argumenty, tzn. zbiory wszystkich wartości występujących w danej kolumnie tabeli.

### Przykład


Wynikiem operacji dopełnienia jest relacja pusta wtw.  $R = U$ . Liczba elementów relacji dopełnienia może być relatywnie duża i wynosi  $card(U) - card(R)$ , tzn.  $card(D_1) \cdot card(D_2) \cdot \dots \cdot card(D_n) - card(R)$ .

## Iloczyn kartezjański

Operację iloczynu kartezjańskiego relacji można przenieść bezpośrednio z algebry zbiorów (relacje są zbiorami). Jednak ze względu na specyficzną strukturę elementów tych zbiorów oraz przyjęty sposób reprezentacji w tabelach wygodnie jest posłużyć się schematami relacji. Iloczyn kartezjański definiuje się dla *dowolnych* relacji (nie ma wymagania zgodności relacji).

Niech  $R(A_1, A_2, \dots, A_n)$  oraz  $S(B_1, B_2, \dots, B_m)$  będą dowolnymi relacjami,  $R \subseteq U$ ,  $S \subseteq V$ . Iloczyn kartezjański relacji  $T = R \times S$  definiowany jest jako relacja o schemacie  $T(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$  taka, że:

$$R \times S = \{(d_1, \dots, d_n, d_{n+1}, \dots, d_{n+m}) : (d_1, \dots, d_n) \in R \wedge (d_{n+1}, \dots, d_{n+m}) \in S\}.$$

Iloczyn kartezjański relacji  $R \times S$  można też zadać intensjonalnie w  $U \times V$  poprzez zdefiniowanie predykatu relacji jak następuje:

$$\|T(d_1, \dots, d_n, d_{n+1}, \dots, d_{n+m})\| = \|R(d_1, \dots, d_n)\| \wedge \|S(d_{n+1}, \dots, d_{n+m})\|.$$

### Przykład

A	B	C	×	B	C	D	=	A	B	C	B	C	D
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>		b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>		a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>		b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>		a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>		b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>		a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>		b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>		a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>		b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>		a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>		b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>		a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>	b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>		b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>		a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>		b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>		a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>
a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>		b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>		a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>	d <sub>3</sub>

Liczba elementów relacji wynikowej  $T = R \times S$  spełnia warunek  $\text{card}(T) = \text{card}(R) \cdot \text{card}(S)$ . Uwaga: operacja iloczynu kartezjańskiego prowadzi zwykle do gwałtownego wzrostu liczby rekordów, spowolnienia operacji i produkcji powtarzającej się informacji!

## Projekcja

*Projekcja (rzutowanie)* jest jedną z podstawowych operacji algebry relacji. Dla intuicji, projekcja oznacza rzutowanie na wybrane kolumny, a więc ograniczenie informacji zadawanej w tabeli reprezentującej relację do wybranych kolumn. Innymi słowy, wybrane kolumny tabeli są zachowane, a inne – usuwane (pomijane; czasowo lub trwale, jeżeli wynik będzie zapisany).

Niech  $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$  będzie wybranym zbiorem atrybutów, a  $R$  będzie relacją o schemacie  $R(A_1, A_2, \dots, A_n)$ .

Niech  $\mathbf{A}' = \{A^1, A^2, \dots, A^k\}$  będzie zbiorem atrybutów wybranych ze zbioru  $\mathbf{A}$ ,  $\mathbf{A}' \subseteq \mathbf{A}$  (ograniczona liczba atrybutów i zmieniona ew. kolejność).

Niech  $P$  będzie relacją o schemacie  $P(A^1, A^2, \dots, A^k)$ .

Operację projekcji  $\pi_P$  definiujemy jako operację postaci  $\pi_P : R \longrightarrow P$  przekształcającą relację  $R$  w pewną relację  $P$ , taką, że:

$$\pi_P((d_1, d_2, \dots, d_n)) = (d^1, d^2, \dots, d^k),$$

gdzie, że  $d^i = d_j$  dla  $A^i = A_j$ ,  $i = 1, 2, \dots, k$ ,  $j \in \{1, 2, \dots, n\}$ ;  $(d_1, d_2, \dots, d_n) \in R$ ,  $(d^1, d^2, \dots, d^k) \in P$ .

### Oznaczenia:

Dla reprezentacji operacji projekcji relacji  $R$  zdefiniowanej jak wyżej stosowane są następujące oznaczenia:

- $\pi_{A^1, A^2, \dots, A^k}(R)$  – rzut relacji  $R$  na atrybuty  $A^1, A^2, \dots, A^k$ ,
- $\pi_{i_1, i_2, \dots, i_k}(R)$  – rzut relacji  $R$  na składowe  $i_1, i_2, \dots, i_k$ ,
- $R[A^1, A^2, \dots, A^k]$  – rzut relacji  $R$  na atrybuty  $A^1, A^2, \dots, A^k$ .

Przykładowo, niech  $R$  będzie relacją o schemacie  $R(A, B, C, D, E)$ ; wówczas

$$\pi_{3,5,1}(R) = \pi_{C,E,A}(R) = R[C, E, A]$$

Logicznie  $\|R(d_1, d_2, \dots, d_n)[A^1, A^2, \dots, A^k]\| \quad (\|R(d^1, d^2, \dots, d^k)\|)$

wtw.  $\exists x_1, x_2, \dots, x_{n-k} R(x_1, \dots, x_{i_1}, d^1, x_{i_1+1}, \dots, x_{i_2}, d^2, \dots, x_{i_k}, d^k, \dots, x_{n-k})$ .



## Projekcja: przykład, własności, problemy

**Przykład:** Niech  $R$  będzie relacją o schemacie  $R(A, B, C, D, E, F)$ :

$A$	$B$	$C$	$D$	$E$	$F$
$a_1$	$b_1$	$c_1$	$d_1$	$e_1$	$f_1$
$a_1$	$b_1$	$c_1$	$d_2$	$e_2$	$f_2$
$a_1$	$b_1$	$c_1$	$d_3$	$e_3$	$f_3$
$a_2$	$b_2$	$c_2$	$d_1$	$e_1$	$f_1$
$a_2$	$b_2$	$c_2$	$d_2$	$e_2$	$f_2$

Projekcją (rzutem) relacji  $R$  na atrybuty  $E, C, A$  jest relacja  $\pi_{E,C,A}(R)$  ( $\pi_{5,3,1}(R)$ ,  $R[E, C, A]$ ) równa:

$E$	$C$	$A$
$e_1$	$c_1$	$a_1$
$e_2$	$c_1$	$a_1$
$e_3$	$c_1$	$a_1$
$e_1$	$c_2$	$a_2$
$e_2$	$c_2$	$a_2$

Własności operacji projekcji:

- zmienia schemat relacji (ogranicza liczbę atrybutów i ich kolejność,
- tracona jest informacja o wartościach niektórych atrybutów,
- mogą powstać duplikaty (rekordy nierozróżnialne),
- liczba rekordów nie zmienia się (o ile nie usuniemy duplikatów ( opcja `DISTINCT` w klauzuli `SELECT`; wówczas liczba rekordów maleje).

Operacja projekcji może prowadzić do utraty możliwości rozróżniania niektórych rekordów; może ona stanowić pewien operator *abstrakcji* (stosowany celowo); ale może też prowadzić do niezamierzonej utraty informacji.

## Antyprojekcja

Antyprojekcja jest stosunkowo rzadko spotykaną operacją algebry relacji. Niech  $R$  będzie relacją o schemacie  $R(A_1, A_2, \dots, A_n, B)$ , gdzie  $B$  jest pewnym wyróżnionym atrybutem (może on występować na dowolnej pozycji).

Niech  $D_1, D_2, \dots, D_n$  będą dziedzinami kolejnych atrybutów  $A_1, A_2, \dots, A_n$ , tzn.:

$$D_i = \pi_i(R),$$

oraz niech  $U = D_1 \times D_2 \times \dots \times D_n$  będzie uniwersum względem atrybutów  $A_1, A_2, \dots, A_n$ .

Niech  $D_B = \pi_B(R)$  będzie zbiorem wszystkich wartości atrybutu  $B$  występujących w relacji  $R$ .

Operacja antyprojekcji relacji  $R$  której wynikiem jest relacja  $R]B[$  (antyprojekcja  $R$  na atrybut  $B$ ) jest zdefiniowana jako:

$$R]B[ = \{b \in B : \forall (d_1, d_2, \dots, d_n) \in U, (d_1, d_2, \dots, d_n, b) \in R\}.$$

**Przykład:** Niech  $R(A_1, A_2, B)$  będzie relacją zdefiniowaną jak następuje:

$A_1$	$A_2$	$B$
$a_1$	$a_1$	$b_1$
$a_2$	$a_1$	$b_1$
$a_1$	$a_2$	$b_1$
$a_2$	$a_2$	$b_1$
$a_1$	$a_1$	$b_2$
$a_2$	$a_1$	$b_2$
$a_1$	$a_2$	$b_2$

Wynikiem operacji antyprojekcji  $R$  na  $B$  jest relacja  $R]B[$  jak poniżej:

$B$
$b_1$

Logicznie  $\|R]B[(b)\|$  wtw. gdy  $\forall (d_1, d_2, \dots, d_n) \in U, (d_1, d_2, \dots, d_n, b) \in R$ .

## Interpretacja operacji antyprojekcji

Operacja antyprojekcji pozwala odpowiadać na pytania typu:

*Którzy dostawcy dostarczają **wszystkie** produkty (występujące w tabeli)?*

**Przykład:** Niech będzie dana tabela dostawców wraz ze specyfikacją dostarczanego produktu:

<i>Dostawca</i>	<i>Element</i>
<i>Budex</i>	<i>cegła</i>
<i>Budex</i>	<i>pustak</i>
<i>Budex</i>	<i>cement</i>
<i>Budex</i>	<i>piasek</i>
<i>Matbud</i>	<i>cegła</i>
<i>Matbud</i>	<i>pustak</i>
<i>Matbud</i>	<i>cement</i>
<i>Matbud</i>	<i>gips</i>
<i>Matbud</i>	<i>piasek</i>
<i>Matbud</i>	<i>żwir</i>
<i>Probud</i>	<i>cegła</i>
<i>Probud</i>	<i>cement</i>
<i>Probud</i>	<i>piasek</i>
<i>Probud</i>	<i>żwir</i>

Wynikiem antyprojekcji tej tabeli na atrybut *Dostawca* jest tabela:

<i>Dostawca</i>
<i>Matbud</i>

Antyprojekcja pozwala odpowiadać na pytania zawierające kwantyfikator ogólny ( $\forall$ ).

## Iloraz

Operacja ilorazu umożliwia *dzielenie* relacji przez relację.

Niech  $R$  będzie relacją o schemacie  $R(A_1, A_2, \dots, A_n)$ , a  $S$  relacją o schemacie  $S(B_1, B_2, \dots, B_k)$ , przy czym zakładamy że  $n > k$  oraz  $S \neq \emptyset$ .

Wynikiem operacji dzielenia relacji  $R$  przez relację  $S$  jest relacja  $R \div S$  będąca zbiorem wszystkich  $(n - k)$ -krotek  $t$ , takich, że dla **wszystkich**  $k$ -krotek  $s \in S$ , krotka  $ts \in R$ .

Niech  $\pi_{1,2,\dots,n-k}(R) = T$ .

$T$  jest zbiorem wszystkich początkowych ciągów krotek relacji  $R$  (o długości  $n - k$ ). Wówczas relacja

$$(T \times S) \setminus R$$

jest zbiorem wszystkich  $n$ -krotek nie należących do  $R$ , utworzonych poprzez konkatencję pierwszych  $n - k$  składowych z relacji  $R$  z **wszystkimi** krotkami z  $S$ .

Niech

$$V = \pi_{1,2,\dots,n-k}((T \times S) \setminus R);$$

$V$  jest zbiorem wszystkich  $n - k$ -krotek  $v$ , takich, że dla pewnej krotki  $s \in S$ , konkatencja  $vs \notin R$ . Zatem

$$T \setminus V = R \div S,$$

lub:

$$R \div S = \pi_{1,2,\dots,n-k}(R) \setminus \pi_{1,2,\dots,n-k}((\pi_{1,2,\dots,n-k}(R) \times S) \setminus R)$$

Operacja dzielenia relacji  $R$  przez relację  $S$  służy do znajdowania wszystkich początkowych ciągów z relacji  $R$  o długości  $n - k$  (wszystkich fragmentów lub początków), które w  $R$  są skonkatencowane ze **wszystkimi** krotkami **zadanymi w relacji  $S$** .

## Iloraz: przykład i interpretacja

Operacja dzielenia relacji  $R$  przez relację  $S$  ma istotne znaczenie praktyczne i stanowi rozszerzenie operacji antyprojekcji. W przypadku operacji antyprojekcji, można było uzyskać odpowiedź na pytanie typu:

*Którzy dostawcy dostarczają **wszystkie** produkty?*

Natomiast w przypadku operacji ilorazu można uzyskać odpowiedź na pytanie typu:

*Którzy dostawcy dostarczają produkty z **określonego** zbioru?*

Interesujący nas zbiór zadawany jest właśnie relacją  $S$ . Poniżej przedstawiono prosty przykład.

**Przykład:**

<i>Dostawca</i>	<i>Element</i>
<i>Budex</i>	<i>cegła</i>
<i>Budex</i>	<i>pustak</i>
<i>Budex</i>	<i>cement</i>
<i>Budex</i>	<i>piasek</i>
<i>Matbud</i>	<i>pustak</i>
<i>Matbud</i>	<i>cement</i>
<i>Matbud</i>	<i>gips</i>
<i>Matbud</i>	<i>piasek</i>
<i>Probud</i>	<i>cegła</i>
<i>Probud</i>	<i>cement</i>
<i>Probud</i>	<i>piasek</i>
<i>Probud</i>	<i>żwir</i>

 $\div$ 

<i>Element</i>
<i>cegła</i>
<i>cement</i>
<i>piasek</i>

 $=$ 

<i>Dostawcy</i>
<i>Budex</i>
<i>Probud</i>

Operacja dzielenia pozwala zatem na realizację zapytań formalizowalnych logicznie z zastosowaniem kwantyfikatora ogólnego ( $\forall$ ).

## Uogólniona suma relacji

Zdefiniujemy uogólnioną sumę relacji (niekoniecznie zgodnych), czyli relacji o różnych schematach. Niech  $R(A_1, A_2, \dots, A_n)$  oraz  $S(B_1, B_2, \dots, B_m)$  będą dowolnymi relacjami,  $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$  oraz  $\mathbf{B} = \{B_1, B_2, \dots, B_m\}$ . Niech  $\mathbf{B}' = \mathbf{B} \setminus \mathbf{A}$ , tzn. zbiór  $\mathbf{B}'$  zawiera wszystkie atrybuty zbioru  $\mathbf{B}$  nie występujące w  $\mathbf{A}$ . Niech  $\mathbf{B}' = \{B^1, B^2, \dots, B^k\}$ ,  $k \leq m$ .

Niech  $R + S$  będzie nową relacją o schemacie zdefiniowanym przez  $(R + S)(A_1, A_2, \dots, A_n, B^1, B^2, \dots, B^k)$ . Relacja  $R + S$  będzie nazywana *uogólnioną sumą* relacji  $R$  i  $S$ , jeżeli dla każdego rekordu  $rs \in R + S$  spełniony jest warunek:

$$\pi_{A_1, A_2, \dots, A_n}(rs) \in R$$

lub

$$\pi_{B_1, B_2, \dots, B_m}(rs) \in S$$

**Przykład:**

A	B	+	B	C	=	A	B	C
a <sub>1</sub>	b <sub>1</sub>		b <sub>1</sub>	c <sub>1</sub>		a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>		b <sub>2</sub>	c <sub>2</sub>		a <sub>1</sub>	b <sub>1</sub>	c <sub>2</sub>
a <sub>3</sub>	b <sub>3</sub>		b <sub>3</sub>	c <sub>3</sub>		a <sub>1</sub>	b <sub>1</sub>	c <sub>3</sub>
						a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
						a <sub>1</sub>	b <sub>3</sub>	c <sub>3</sub>
						a <sub>2</sub>	b <sub>1</sub>	c <sub>1</sub>
						a <sub>2</sub>	b <sub>2</sub>	c <sub>1</sub>
						a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>
						a <sub>2</sub>	b <sub>2</sub>	c <sub>3</sub>
						a <sub>2</sub>	b <sub>3</sub>	c <sub>3</sub>
						a <sub>3</sub>	b <sub>1</sub>	c <sub>1</sub>
						a <sub>3</sub>	b <sub>2</sub>	c <sub>2</sub>
						a <sub>3</sub>	b <sub>3</sub>	c <sub>1</sub>
						a <sub>3</sub>	b <sub>3</sub>	c <sub>2</sub>
						a <sub>3</sub>	b <sub>3</sub>	c <sub>3</sub>

## Interpretacja sumy uogólnionej: przykład

Uogólniona suma relacji jest relacją o schemacie zawierającym wszystkie atrybuty obu relacji (powtarzające się atrybuty występują tylko raz) oraz wszystkie krotki zbudowane z elementów relacji  $R$  i  $S$ , takie, że obcięcie danej krotki do zbioru atrybutów relacji  $R$  należy do tej relacji lub obcięcie do zbioru atrybutów relacji  $S$  należy do tej relacji. Uogólniona suma relacji odpowiada na pytanie typu:

*Jakich odbiorców mogłyby (potencjalnie) znaleźć produkty dostarczane przez pewnych dostawców oraz którzy dostawcy mogliby (potencjalnie) dotarczać produkty dla pewnych odbiorców?*

**Przykład:**

<i>Dostawca</i>	<i>Produkt</i>		<i>Produkt</i>	<i>Odbiorca</i>		<i>Dostawca</i>	<i>Produkt</i>	<i>Odbiorca</i>
<i>Budex</i>	<i>cegła</i>	+	<i>cegła</i>	<i>Wykbud</i>	=	<i>Budex</i>	<i>cegła</i>	<i>Rembud</i>
<i>Budex</i>	<i>pustak</i>		<i>pustak</i>	<i>Wykbud</i>		<i>Budex</i>	<i>pustak</i>	<i>Rembud</i>
<i>Budex</i>	<i>cement</i>		<i>cement</i>	<i>Rembud</i>		<i>Budex</i>	<i>cement</i>	<i>Rembud</i>
<i>Budex</i>	<i>piasek</i>		<i>piasek</i>	<i>Wykbud</i>		<i>Budex</i>	<i>piasek</i>	<i>Rembud</i>
<i>Matbud</i>	<i>pustak</i>		<i>piasek</i>	<i>Rembud</i>		<i>Matbud</i>	<i>pustak</i>	<i>Rembud</i>
<i>Matbud</i>	<i>cement</i>		<i>gips</i>	<i>Wykbud</i>		<i>Budex</i>	<i>cegła</i>	<i>Wykbud</i>
<i>Matbud</i>	<i>gips</i>		<i>gips</i>	<i>Rembud</i>		<i>Budex</i>	<i>pustak</i>	<i>Wykbud</i>
<i>Matbud</i>	<i>piasek</i>					<i>Budex</i>	<i>cement</i>	<i>Wykbud</i>
				<i>Budex</i>	<i>piasek</i>	<i>Wykbud</i>		
				<i>Matbud</i>	<i>pustak</i>	<i>Wykbud</i>		
				<i>Matbud</i>	<i>cement</i>	<i>Wykbud</i>		
				<i>Matbud</i>	<i>gips</i>	<i>Wykbud</i>		
				<i>Matbud</i>	<i>piasek</i>	<i>Wykbud</i>		

## Wyszukiwanie danych: operacja selekcji

Operacja **selekcji** jest podstawową operacją realizowaną w relacyjnych bazach danych. Jest ona stosowana do wyszukiwania danych spełniających zadane *kryteria* (warunki). Operację selekcji można więc traktować jako sposób *inetynsjonalnego* definiowania danych (interesującej nas relacji) w uniwersum zadawanym *ekstensjonalnie* przez relację wyjściową.

Niech  $R$  będzie relacją o schemacie  $R(A_1, A_2, \dots, A_n)$ . Zauważmy, że atrybuty  $A_1, A_2, \dots, A_n$  można traktować jak *zmiennie*, które dla poszczególnych rekordów przybierają określone wartości atomiczne. Niech  $\Psi$  oznacza pewną formułę logiczną zbudowaną ze:

- stałych (w tym liczb i łańcuchów znakowych),
- atrybutów  $A_1, A_2, \dots, A_n$  pełniących rolę zmiennych; w formule  $\Phi$  są one zmiennymi wolnymi (stosowana notacja:  $[A_i]$ ),
- funkcji wbudowanych w systemie (określonych na stałych, zmiennych oraz wartościach funkcji),
- symboli relacji wbudowanych w systemie (np.  $<$ ,  $>=$ , etc.),
- spójników logicznych dopuszczalnych w systemie, oraz
- nawiasów definiujących kolejność operacji.

Formuła  $\Psi$  definiuje **kryterium selekcji**. Poprawnie zdefiniowana formuła  $\Psi$  definiuje jednoznacznie relację  $R_\Psi$ , taką, że:

$$R_\Psi = \{(d_1, d_2, \dots, d_n) \in R : \Phi\{A_1/d_1, A_2/d_2, \dots, A_n/d_n\}\}. \quad (1)$$

Zapis  $\{A_1/d_1, A_2/d_2, \dots, A_n/d_n\}$  oznacza podstawienie stałych  $d_1, d_2, \dots, d_n$  za zmienne  $A_1, A_2, \dots, A_n$ . Operacja selekcji definiowana jest poprzez zadanie formuły  $\Psi$  a jej wynik poprzez równość (1).



## Operacja selekcji: definiowanie i własności

---

---

Z formułą  $\Psi$  można jednoznacznie związać *funkcję selekcji*  $\sigma_\Psi$ , określoną jak następuje:

$$\sigma_\Psi : R \longrightarrow R_\Psi.$$

Funkcja selekcji (wyboru) może być także interpretowana jako funkcja charakterystyczna poszukiwanej relacji, tzn.:

$$\sigma_\Psi(d_1, d_2, \dots, d_n) = \begin{cases} 1 & \text{jeżeli } (d_1, d_2, \dots, d_n) \in R_\Psi \\ 0 & \text{jeżeli } (d_1, d_2, \dots, d_n) \notin R_\Psi \end{cases}$$

Tak więc przy zadanej wyjściowej relacji  $R$  oraz warunku selekcji  $\Psi$ , operacja selekcji może być zapisana w postaci:

$$\sigma_\Psi(R) = R_\Psi,$$

gdzie  $R_\Psi$  jest zbiorem tych wszystkich  $n$ -krotek relacji  $R$ , które spełniają warunek  $\Psi$ ,  $R_\Psi \subseteq R$ .

W postaci logicznej operację selekcji można też zdefiniować jako:

$$\|\sigma_\Psi(R)\| = \|R\| \wedge \|\Psi\|.$$

Własności operacji selekcji:

- zostaje zachowany schemat relacji, oraz
- w odniesieniu do wyselekcjonowanych krotek nie jest tracona informacja (o ile jednocześnie nie jest zastosowana operacja projekcji, zostaje zachowany ich pełny opis),
- zazwyczaj znacznemu zmniejszeniu ulega liczba rekordów (w odniesieniu do relacji wyjściowej),
- wynikiem operacji selekcji może być zbiór zawierający 0, 1, lub wiele krotek relacji wyjściowej.

## Przykład operacji selekcji

Rozważmy relację  $R$  zadaną tablicą Pracownik:

ID_prac	Nazwisko	Imię	Data ur	Stanowisko	Dział	Stawka
MT101	Abacki	Adam	61-01-01	robotnik	P10	550,00 zł
MT102	Abakowski	Alojzy	61-01-02	robotnik	P10	574, 00 zł
MT103	Adamski	Antoni	61-01-03	robotnik	P20	1275,00 zł
MT104	Adamski	Arnold	61-01-03	robotnik	P20	1280,00 zł
MT105	Adamski	Arnold	61-01-03	robotnik	P20	1295,00 zł
KT101	Aron	Antonina	61-01-03	robotnik	P10	575,00 zł
MU101	Batman	Bogusław	67-02-13	kierownik	P30	1224,00 zł
KU101	Celińska	Mirosława	69-03-08	analityk	F10	975,00 zł
MV101	Dioniziak	Dariusz	71-10-17	v-prezes	V	3000,00 zł

Niech  $\Psi_1 = \text{Nazwisko} = \text{'Adamski'}$ ; wówczas  $\sigma_1(R)$  jest tablicą postaci:

ID_prac	Nazwisko	Imię	Data ur	Stanowisko	Dział	Stawka
MT103	Adamski	Antoni	61-01-03	robotnik	P20	1275,00 zł
MT104	Adamski	Arnold	61-01-03	robotnik	P20	1280,00 zł
MT105	Adamski	Arnold	61-01-03	robotnik	P20	1295,00 zł

Podobnie,  $\Psi_2 = (\text{Stanowisko} = \text{'robotnik'} \text{ AND } \text{Stawka} < 1000,00 \text{ zł}) \text{ OR } (\text{Stanowisko} = \text{'analityk'} \text{ AND } \text{Stawka} < 1000,00 \text{ zł})$  da wynik  $\sigma_2(R)$ :

ID_prac	Nazwisko	Imię	Data ur	Stanowisko	Dział	Stawka
MT101	Abacki	Adam	61-01-01	robotnik	P10	550,00 zł
MT102	Abakowski	Alojzy	61-01-02	robotnik	P10	574, 00 zł
KT101	Aron	Antonina	61-01-03	robotnik	P10	575,00 zł
KU101	Celińska	Mirosława	69-03-08	analityk	F10	975,00 zł

---

---

## Definiowanie warunku selekcji

---

---

Z logicznego punktu widzenia, warunek selekcji  $\Psi$  musi być *poprawnie zdefiniowaną formułą*, której wartość logiczna jest określona dla wszystkich rekordów tabeli wyjściowej. W praktyce, definiowanie kryterium selekcji odbywa się z wykorzystaniem:

- predefiniowanego szablonu użytkownika,
- właściwego dla systemu szablonu *Query by Example*,
- języka SQL (instrukcji wprowadzonej ręcznie, generowanej automatycznie lub z poziomu języka wyższego rzędu),
- specyficznego dla systemu języka zapytań (np. VTLIS, Datalog, Prolog).

Typowe elementy definiowania warunku selekcji obejmują:

- użycie stałych:
  - numerycznych, np. 13, +13, -13, 13.57 (13,57),
  - tekstowych, np. 'Kowalski', "Kowalski",
  - logicznych, *True*, *False* (-1, 0) lub (0,1),
  - datowych, np. #99-08-13#, '1999-08-13', '1999/08/13',
- nazw atrybutów i wyrażeń ścieżkowych, np. Nazwisko, [Data ur], Pracownik.Nazwisko,
- operatorów:
  - arytmetycznych, np. +, -, \*, /,
  - tekstowych, np. +, &, %, \_
  - logicznych, np. *and*, *or*, *not*, *xor*,
  - relacyjnych, np. =, <, >, <=, >=, <>, !=,
  - specjalnych (porównania), np. *Like*, *Between*,
- funkcji, np. *Abs(.)*, *Sqr(.)*, *Trim(.)*, itp.

## Idea szablonu QBE

---



---

### Definiowanie szablonu QBE

Aby zdefiniować szablon zapytania używając języka QBE należy określić:

- **Atrybuty** – zdefiniować, które pola mają być uwzględnione w definiowanym zapytaniu; podaje się też ich kolejność,
- **Źródło atrybutu** – tabela lub zapytanie z którego pochodzi dany atrybut (uwaga: w zapytaniu mogą występować atrybuty o takich samych nazwach, ale pochodzące z różnych źródeł),
- **Sortowanie** – czy i jak wyniki zapytania mają być posortowane,
- **Wyświetlanie** – czy dane pole ma być wyświetlane,
- **Kryterium selekcji** – formuły definiujące warunek selekcji, a więc *intensjonalną* definicję relacji docelowej.

### Schemat szablonu QBE

<i>Pole :</i>	$A_1$	$A_2$	...	$A_j$	...	$A_k$
<i>Tabela :</i>	$T_1$	$T_2$	...	$T_j$	...	$T_k$
<i>Sortuj :</i>			...	$r/m/b$	...	
<i>Pokaż :</i>			...	$y/n$	...	
<i>Kryteria :</i>	$k_1$	$k_2$	...	$k_j$	...	$k_k$
<i>Lub :</i>	$k'_1$	$k'_2$	...	$k'_j$	...	$k'_k$

(2)

W wyniku działania kwerendy otrzymuje się tabelę o schemacie  $K(A_1, A_2, \dots, A_k)$ ; w zależności od tego, ile rekordów spełnia określone kryteria, w tabeli wynikowej może być 0, 1, lub wiele rekordów. Mogą one podlegać dalszemu przetwarzaniu.

---

## System ACCESS – użycie wyrażeń

---

### Definiowanie wyrażeń

Wyrażenia to definicje funkcji określające sposób wyliczania pewnych wartości; mogą one być konstruowane za pomocą *stałych*, atrybutów (pełniących rolę zmiennych), operatorów (arytmetycznych, logicznych, etc.) oraz predefiniowanych (w systemie) funkcji.

Stałe obejmują:

- **stałe numeryczne**, np. 13, +13, -13, +13, 34, 13.35,
- **stałe tekstowe**, np. "Kowalski", "Kraków", "jan kowalski",
- **stałe logiczne** – PRAWDA/True oraz FAŁSZ/False; liczbowymi odpowiednikami są -1 dla prawdy oraz 0 dla fałszu,
- **stałe datowe** – określają wartość daty i występują w ogranicznikach #, np. #07-22-98#.

Operatory obejmują:

- **operatory arytmetyczne** – dodawanie (+), odejmowanie (-), mnożenie (\*), dzielenie (/), potęgowanie (^),
- **operatory relacyjne** – =, >, <, <=, >=, <>, BETWEEN ... AND ..., IS NULL, IS NOT NULL, ich wynik jest wartością logiczną (PRAWDA lub FAŁSZ); mogą być stosowane do porównywania liczb (dat, walut) i tekstów,
- **operatory tekstowe** – +, &, są to operatory konkatencji tekstów,
- **operatory datowe** – odejmowanie dat (-) (wynik w dniach), modyfikacja daty (+, -), wynikiem jest data,
- **operatory logiczne** – koniunkcja (AND), alternatywa (OR) i negacja (NOT).

---

## System ACCESS – typy kwerend

---

### Typy kwerend w systemie ACCESS

- **Kwerenda wybierająca.** Kwerenda taka zwraca dane z jednej lub wielu tabel. Realizuje operacje projekcji i selekcji; dodatkowo może dostarczać pól wyliczeniowych. Pozwala uzyskać efekt *perspektywy*; dane zmodyfikowane w tabeli będącej wynikiem zapytania zostaną zmodyfikowane (bez ostrzeżenia!) w tabelach wyjściowych. Pozwala również na agregację danych.
- **Kwerenda krzyżowa** – specjalny typ kwerendy wyszukującej i jednocześnie grupującej dane wg specyficznych wartości z wybranej kolumny; wartości te stają się nagłówkami kolumn (transpozycja),
- **Kwerendy funkcjonalne.** Realizują różne operacje prowadzące do *trwałych zmian* w tabelach.
  - **Kwerenda tworząca tabelę** (na podstawie innej tabeli),
  - **Kwerenda usuwająca** – usuwa dane z tabeli,
  - **Kwerenda dołączająca** - dołącza dane z tabeli do innej tabeli,
  - **Kwerenda aktualizująca** – modyfikuje dane w tabeli.
- **Kwerendy SQL**
  - **Kwerenda składająca** – realizuje sumę tabel,
  - **Kwerenda przekazująca** – przekazuje instrukcję SQL do serwera zewnętrznego,
  - **Kwerenda definiująca dane** – pozwala definiować tabele.
- **Kwerenda parametryczna** – pozwala tworzyć kwerendy dynamicznie aktualizujące parametry zapytania z wykorzystaniem okna dialogowego (argument definiowany jest w postaci [zapytanie o argument]).

---

## System ACCESS – funkcje wbudowane

---

---

### Funkcje arytmetyczne

- $\text{Exp}(n)$  – oblicza  $e$  podniesione do potęgi  $n$ ,
- $\text{Log}(n)$  – oblicza logarytm naturalny z  $n$ ,
- $\text{Sqr}(n)$  – oblicza pierwiastek kwadratowy z  $n$ ,
- $\text{Int}(n)$  – oblicza część całkowitą argumentu;  $\text{Int}(n) \leq n$ ,
- $\text{Fix}(n)$  – oblicza część całkowitą argumentu;  $\text{Fix}(n) \leq |n|$ ,
- $\text{Abs}(n)$  – oblicza wartość bezwzględną z  $n$ ,
- $\text{Sgn}(n)$  – zwraca znak liczby (+1, -1 lub 0).

### Funkcje konwersji

- $\text{Chr}(n)$  – zamienia wartość  $n$  na odpowiadający znak ASCII,
- $\text{Asc}(c)$  – zwraca kod ASCII znaku  $c$ ,
- $\text{Str}(n)$  – zamienia liczbę  $n$  na łańcuch znaków,
- $\text{Val}(c)$  – zamienia łańcuch znaków (zgodny z formatem liczby!) na tę liczbę,
- $\text{Clng}(w)$  – zamienia argument na liczbę całkowitą długą,
- $\text{DateSerial}(\text{rok}; \text{miesiąc}; \text{dzień})$  – wyznacza datę dla podanych argumentów,
- $\text{Day}(n)$ ,  $\text{Month}(n)$ ,  $\text{Weekday}(n)$ ,  $\text{Year}(n)$  – wyznacza element daty odpowiadający liczbie  $n$ .

---

## System ACCESS – funkcje wbudowane

---

---

### Funkcje operujące na łańcuchach znaków

- `Space(n)` – zwraca ciąg spacji o długości  $n$ ,
- `String(n; c)` – zwraca ciąg  $n$  powtórzeń pierwszego znaku  $c$ ,
- `InStr(n; c1; c2; jak)` – wyznacza pozycje wystąpienia podłańcucha  $c2$  w  $c1$  począwszy od znaku  $n$ ; parametr  $jak$  ustala, czy uwzględniania jest wielkość liter ( $jak=0,1,2$ ),
- `Left(c; n)` – zwraca  $n$  pierwszych od lewej znaków łańcucha  $c$ ,
- `Right(c; n)` – zwraca  $n$  znaków łańcucha  $c$  od prawej,
- `Mid(c; n1; n2)` – zwraca  $n2$  znaków łańcucha  $c$  począwszy od znaku  $n1$ ,
- `LTrim(c)`, `RTrim(c)`, `Trim(c)` – usuwa spacje wiodące, spacje z prawej strony lub obustronne,
- `LCase(c)` – zamienia litery łańcucha  $c$  na małe,
- `UCase(c)` – zamienia litery łańcucha  $c$  na duże,
- `Len(c)` – zwraca długość łańcucha  $c$ ,
- `StrComp(c1; c2; jak)` – zwraca  $-1$  lub  $1$  w zależności od tego czy łańcuch  $c1$  poprzedza  $c2$  czy odwrotnie; zwraca  $0$  dla identycznych łańcuchów. Parametr  $jak = 0,1,2$  definiuje sposób porównywania ( $0$  – uwzględnia wielkość liter,  $1$  – nie uwzględnia wielkości liter,  $2$  – zgodnie z ustawieniami systemowymi bazy danych).



---

## System ACCESS – funkcje wbudowane

---

### Funkcje daty i czasu

- `Date()`, `Time()`, `Now()` – zwracają dane z zegara systemowego (datę, czas, datę i czas),

### Funkcje sprawdzające

- `IsDate(arg)`, `IsEmpty(arg)`, `IsNull(arg)`, `IsNumeric(arg)` – sprawdzają, czy podany argument jest odpowiednio: datą, łańcuchem pustym, ma wartość *Null*, oraz czy jest liczbą.

### Funkcja wyboru

- `Iif(l; w1; w2)` – jeżeli wyrażenie logiczne *l* jest prawdziwe, zwraca *w1*; w przeciwnym przypadku zwraca *w2*. Przykład zastosowania: `Iif(IsNull([Cena]); "Brak ceny"; Str([Cena])+"PLN")`.

### Operator porównania stringów Like

- `Like "wzorzec"` – porównuje poprzedzający argumen ze wzorcem; we wzorcu można użyć symboli wieloznacznych,
- `*` – dowolny ciąg znaków, również pusty,
- `?` – dokładnie jeden znak,
- `[]` – dowolny ze znaków w nawiasach (`[xyz]` – *x*, *y* albo *z*),
- `[!]` – żaden ze znaków po `!"` (`[!xyz]` – nie *x*, ani *y* ani *z*),
- `[a-z]` – znak z zakresu od *a* do *z* (`[0-9]` – dowolna cyfra),
- `#` – dowolna cyfra.