

Operatory arytmetyczne (tab. 6.1) służą do obliczania wartości z dwóch lub więcej liczb albo do zmiany znaku liczby z plus na minus.

Tab. 6.1. Operatory arytmetyczne

Operator	Przeznaczenie	Przykład
+	Suma dwóch liczb	[Wpłata] + [Dodatek]
-	Różnica dwóch liczb lub oznaczenie ujemnej wartości liczby	[Wpłata] - [Rabat]
*	Iloczyn dwóch liczb	[Wpłata]*0.25
/	Iloraz dwóch liczb	[Suma]/[Liczba]
\	Zaokrąglenie obu liczb do wartości całkowitych, a następnie podzielenie pierwszej liczby przez drugą i obcięcie wyniku do liczby całkowitej	[Kwota]\[Liczba]
Mod	Reszta z dzielenia pierwszej liczby przez drugą	[Zakwaterowane] Mod [Pokoje]
^	Podniesienie liczby do potęgi podanej w wykładniku	Liczba ^ Wykładnik

Operatory porównywania (tab. 6.2) służą do porównywania wartości i zwracania wyniku w postaci „prawda”, „fałsz” lub „Null”.

Tab. 6.2. Operatory porównywania

Operator	Przeznaczenie
<	Określenie, czy pierwsza wartość jest mniejsza niż druga
<=	Określenie, czy pierwsza wartość jest mniejsza niż druga lub jej równa
>	Określenie, czy pierwsza wartość jest większa niż druga
>=	Określenie, czy pierwsza wartość jest większa niż druga lub jej równa
=	Określenie, czy pierwsza wartość jest równa drugiej
<>	Określenie, czy pierwsza wartość nie jest równa drugiej

We wszystkich przypadkach, jeśli jedna wartość jest pusta (*Null*), wynik jest również pusty (*Null*). Ponieważ wartość pusta oznacza wartość nieznana, wynik porównania z taką wartością jest również nieznany.

Operatory logiczne (tab. 6.3) służą do łączenia dwóch wartości i zwracania wyniku w postaci „prawda”, „fałsz” lub „Null”.

Tab. 6.3. Operatory logiczne

Operator	Zastosowanie	Opis
And	Wyr1 And Wyr2	Zwraca wartość „prawda”, jeśli Wyr1 i Wyr2 mają wartość „prawda”
Or	Wyr1 Or Wyr2	Zwraca wartość „prawda”, jeśli Wyr1 lub Wyr2 ma wartość „prawda”
Eqv	Wyr1 Eqv Wyr2	Zwraca wartość „prawda”, jeśli Wyr1 i Wyr2 mają wartość „prawda” lub Wyr1 i Wyr2 mają wartość „fałsz”
Not	Not Wyr	Zwraca wartość „prawda”, jeśli Wyr nie jest „prawda” lub zwraca „fałsz”, gdy Wyr jest „prawda”
Xor	Wyr1 Xor Wyr2	Zwraca wartość „prawda”, jeśli jedno z wyrażeń Wyr1 lub Wyr2 ma wartość „prawda”, ale nie oba

Operatory łączenia tekstów (tab. 6.4) służą do sklejania dwóch wartości tekstowych w jedną.

Tab. 6.4. Operatory łączenia tekstów

Operator	Zastosowanie	Opis
&	ciąg1 & ciąg2	Łączy dwa ciągi w jeden
+	ciąg1 + ciąg2	Łączy dwa ciągi w jeden, informując o wartościach pustych

Operatory specjalne przedstawiono w tabeli 6.5.

Tab. 6.5. Operatory specjalne

Operator	Opis
<i>Is Null</i> lub <i>Is Not Null</i>	Określa, czy wartość jest pusta (<i>Null</i>), czy też nie jest pusta (<i>Is Not Null</i>)
<i>Like</i> "wzorzec"	Określa zgodność wartości i „wzorca” z uwzględnieniem symboli wieloznacznych we „wzorcu”.
<i>Between</i> wartość1 <i>And</i> wartość2	Określa, czy wartość liczbowa lub wartość daty mieści się w określonym zakresie
<i>In</i> (ciąg1, ciąg2...)	Określa, czy dana wartość należy do przedstawionego zbioru wartości

Przykłady wyrażeń przedstawiono także w tabelach 6.6 - 6.8. Zamieszczone w nich wzory wyrażeń będą użyteczne przy wykonywaniu następnych zadań z tego i z kolejnych rozdziałów.

Przykłady kryteriów dla pól liczbowych

Przykłady wyrażeń zbudowanych na danych liczbowych przedstawiono w tabeli 6.6.

Tab. 6.6. Przykłady wyrażeń dla danych liczbowych

Przykład wyrażenia dla danych liczbowych	Objaśnienie
120	Wartość jest równa 120
= 120	
<> 120	Wartość nie jest równa 120
Not 120	
> 120	Wartość jest większa od 120
< 120	Wartość jest mniejsza od 120
>= 120	Wartość jest większa lub równa 120
<= 120	Wartość jest mniejsza lub równa 120
120 Or 220	Wartość jest równa 120 lub 220
In (120; 220)	
(<>120) And (<>200)	Wartość jest nie równa 120 ani 220
Not(120 Or 220)	
<>(120 Or 220)	
>= 120 And <= 220	Wartość jest z zakresu <120; 220>, tzn. jest większa lub równa 120 i jednocześnie jest mniejsza lub równa 220
Between 120 And 220	
> 120 And < 220	Wartość jest większa od 120 oraz mniejsza od 220
< 120 Or > 220	Wartość jest mniejsza od 120 albo większa od 220 (tzn. jest spoza zakresu <120; 220>)
Not(>= 120 And <= 220)	
< 120 Or >= 220	Wartość jest mniejsza od 120 albo równa lub większa od 220
<= 120 Or >220	Wartość jest mniejsza lub równa 120 albo większa od 220
<= 120 Or 220	Wartość jest równa lub mniejsza od 120 albo równa 220
>= 120 And < 220	Wartość jest większa lub równa 120 oraz mniejsza od 220
> 120 And <= 220	Wartość jest większa od 120 oraz mniejsza lub równa 220

Przykłady kryteriów dla pól zawierających daty

W wyrażeniu, które posłuży jako kryterium filtrowania pól zawierających daty, mogą być użyte operatory porównania, logiczne i specjalne. W tabeli 6.7 przedstawiono przykłady takich wyrażeń.

Zapis daty w środowisku programu **Access** różni się od danych innych typów tym, że każda data w wyrażeniu powinna zaczynać się i kończyć znakiem #. Między znakiem # i liczbą nie wolno używać spacji.

Tab. 6.7. Przykłady wyrażeń dla danych typu **Data/Godzina**

Przykład wyrażenia dla danych typu <i>Data/Godzina</i> (format <i>Data krótka</i>)	Objaśnienie
#2001-10-25#	Data jest równa 2001-10-25
=#2001-10-25#	
Not #2001-10-25#	Wszystkie daty oprócz dnia 2001-10-25
<>#2001-10-25#	
< #1996-1-1#	Daty sprzed roku 1996
<= #1995-12-31#	
>= #1991-01-01#	Daty po roku 1991
> #1990-12-31#	
>= #2014-10-01# And <=#2015-01-01#	Daty z zakresu czasu <2014-10-01; 2015-01-01>
Between #2014-10-01# And #2015-01-01#	
<#2014-10-01# OR >#2015-01-01#	Daty spoza zakresu czasu <2014-10-01; 2015-01-01>
Not (>=#2014-10-01# AND <=#2015-01-01#)	
Not Between #2014-10-01# And #2015-01-01#	Daty z roku 1997
> #1996-12-31# And < #1998-1-1#	
>= #1997-1-1# And <= #1997-12-31#	
Between #1997-1-1# And #1997-12-31#	
Between #1995-1-5# And #1995-1-10#	Dowolna data z zakresu dni od 5 stycznia 1995 roku do 10 stycznia 1995 roku
>= #1995-1-5# And <= #1995-1-10#	
< #1997-1-1# Or > #2000-12-31#	Dowolna data spoza zakresu lat <1997 ; 2000>
In(#2001-10-12#; #2001-11-22#)	Data jest równa jednej z dwóch: 2001-10-12 lub 2001-11-22 Można zastosować In do do- wolnej skończonej liczby dat
Not In(#2001-10-12#; #2001-11-22#)	Nie jest to data 2001-10-12 ani 2001-11-22

*Przykłady kryteriów dla pól zawierających
dane tekstowe*

Każdą wartość tekstową w wyrażeniu należy ujmować w znaki cudzo-
słowa. W prostych wyrażeniach program dodaje je automatycznie.

Możemy używać w wyrażeniu także symboli wieloznacznych.
Są to: * (**gwiazdka**) oraz ? (**znak zapytania**).

Symbol wieloznaczny * w wyrażeniu oznacza dowolną liczbę dowolnych
symboli.

Symbol wieloznaczny ? w wyrażeniu oznacza dowolny pojedynczy
znak.

Aby nie wymieniać kolejnej litery alfabetu, można stosować zapis w postaci **[Litera1–Litera2]**. Taki zapis oznacza kolejne litery alfabetu, zaczynając od **Litery1**, a kończąc **Litera2**. Na przykład wyrażenie **Like "[C-E]"** można wykorzystać do przedstawienia liter **C**, **D** lub **E**.

W tabeli 6.8. przedstawiono przykłady wyrażień dla danych tekstowych.

Tab. 6.8. Przykłady wyrażień dla danych tekstowych

Przykład wyrażenia dla danych tekstowych	Objaśnienie
<i>Like "Adam"</i>	Wartość tekstowa jest równa "Adam"
"Adam"	
<i>Like "Piotr" Or "Marek"</i>	Wartość tekstowa jest równa "Piotr" lub "Marek"
"Piotr" Or "Marek"	
<i>In("Adam"; "Piotr"; "Marek")</i>	Wartość tekstowa jest równa "Adam" lub "Piotr", lub "Marek"
<i>Not In("Adam"; "Piotr"; "Marek")</i>	Wartość tekstowa nie jest równa "Adam" ani "Piotr", ani "Marek"
<i>Not "Mechaniczny"</i>	Dowolny tekst oprócz "Mechaniczny"
<i>Not Like "Mechaniczny"</i>	
<i>Like "D*"</i>	Wartość tekstowa zaczyna się od litery "D"
<i>Not Like "D*"</i>	Wartość tekstowa nie zaczyna się od litery "D"
<i>Not "D*"</i>	
<i>Like "*d"</i>	Wartość tekstowa kończy się literą "D"
<i>Like "An*"</i>	Wartość tekstowa zaczyna się od "An"
<i>Like "A*" Or "D*"</i>	Wartość tekstowa zaczyna się od litery "A" lub od litery "D"
<i>Like "[A-D]*"</i>	Wartość tekstowa zaczyna się od litery "A" lub "B", lub "C", lub "D"
<i>Like "Biał? "</i>	Wartości tekstowe, które mają dokładnie 5 liter, pierwsze 4 to są "Biał", a ostatnia dowolna
<i>Like "K?wał*"</i>	Wartości tekstowe, zaczynające się od litery "K", których druga litera jest dowolna, następnie są litery "wał" i dowolny ciąg znaków.

Prostym przykładem zastosowania operatora **Or** jest filtr utworzony w kolejnym zadaniu.

Zadanie 6_9

Wykonaj filtrowanie tabeli **Klienci** metodą **filtruj według formularza**, umieszczając kryterium na jednej zakładce. Wyświetl rekordy dotyczące właścicieli firm mających na imię „Karolina” lub „Janusz”.