

F8 - filtrowanie przy pomocy wyrażeń

Opcja „Wyszukiwanie tekstowe” ma tryb, w którym można filtrować dane za pomocą specjalnych wyrażeń. Aby skorzystać z tego trybu, nie trzeba nic włączać - wystarczy wpisać w pole wyszukiwania wyrażenie zawierające nazwę jednej z kolumn, ujętą w nawiasy klamrowe, np. `{Status} = "W"`, a następnie zatwierdzić klawiszem Enter. Pozycje na liście zostaną wtedy przefiltrowane w taki sposób, że widoczne będą tylko te, które spełniają warunki podane w wyrażeniu. We wspomnianym przykładzie będą to zatem wiersze, dla których w kolumnie „Status” znajduje się wartość „W”.

Składnia wyrażeń opiera się na znanej z języka C# technologii LINQ, w związku z tym ten sposób pracy z F8 jest polecany użytkownikom posiadającym pewną wiedzę z zakresu programowania.

Filtrowanie przy pomocy wyrażeń nie jest dostępne na listach edycyjnych.

Nazwy kolumn

Gdy w polu wyszukiwania pojawi się znak `{`, program wyświetla listę podpowiedzi z nazwami kolumn. Można z niej wybrać kolumnę, która zostanie wstawiona do wyrażenia. Aby zamknąć listę bez wybierania kolumny, należy wcisnąć klawisz Esc lub Tab. Użycie kombinacji klawiszy Ctrl+kropka sprawi, że lista pokaże się na nowo - pod warunkiem, że karetką w polu wyszukiwania znajduje się w miejscu, gdzie mamy niedokończoną nazwę kolumny.

Nazwy kolumn użyte w wyrażeniu mogą być nazwami pełnymi lub skróconymi. Wyrażenie `{Typ dokumentu} = "FS"` jest zatem równoważne wyrażeniu `{T} = "FS"`.

Wielkość liter w nazwie kolumny nie ma znaczenia, więc `{t}` będzie oznaczało to samo co `{T}`.

Nie jest wymagane także używanie polskich znaków diakrytycznych, więc `{Wartosc}` będzie tym samym co `{Wartość}`.

Porównywanie wartości kolumn

Porównywanie wartości kolumn wygląda różnie w zależności od tego, jaki typ danych zawiera kolumna.

Można porównywać między sobą wartości dwóch kolumn albo porównywać wartość jednej kolumny do podanej stałej.

Przykład porównania między sobą wartości dwóch kolumn:

```
{Data zakończenia dostawy} > {Data wystawienia}
```

Przykład porównania do stałej:

```
{Status} = "W"
```

Kolumny zawierające tekst

Tekst jest równy X lub nie jest równy X

W kolumnach zawierających tekst można porównywać wartość z kolumny przy pomocy operatora `=`. Tekst, do którego porównywana jest wartość, należy ująć w cudzysłów, np:

```
{Status} = "W"
```

Tak podany warunek będzie spełniony, jeśli tekst w kolumnie „Status” będzie dokładnie identyczny z podanym, czyli równy „W”.

Wielkość liter jest istotna, co oznacza, że warunek {Typ dokumentu} = "fs" nie będzie spełniony przez dokumenty o typie „FS”.

Wymagane jest również używanie polskich znaków, więc warunek {Forma płatności} = "Gotówka" nie będzie spełniony przez pozycje, w których forma płatności to „Gotówka”.

Aby uzyskać pozycje, w których tekst w danej kolumnie nie jest równy podanemu tekstowi, należy zanegować warunek równości.

```
{Status} != "W"  
!({Status} = "W")
```

Powyższe warunki są równoznaczne i będą spełnione, jeśli w kolumnie „Status” znajdzie się jakikolwiek inny tekst niż „W”.

Tekst zawiera X lub nie zawiera X

Aby uzyskać pozycje, które zawierają podany tekst, należy skorzystać z funkcji `Contains` języka C#.

```
{Nabywca}.Contains("Drogeria")  
{Nabywca}.Contains("Drogeria") == true
```

Powyższe warunki są równoznaczne i będą spełnione, jeśli w kolumnie „Nabywca” znajdą się teksty zawierające słowo „Drogeria”, czyli np. „Drogeria ALEGRO”, „Drogeria NOVUM” albo „Najlepsza Drogeria”. Wielkość liter ma tu znaczenie, a polskie znaki są wymagane.

Aby uzyskać pozycje, które nie zawierają podanego tekstu, należy użyć takich samych wyrażeń jak powyżej, ale zanegować ich wynik.

```
!{Nabywca}.Contains("Drogeria")  
{Nabywca}.Contains("Drogeria") == false
```

W obu powyższych przypadkach pokazane zostaną tylko pozycje, które nie mają w kolumnie „Nabywca” tekstu „Drogeria”.

Tekst zaczyna się od X albo nie zaczyna się od X

Aby uzyskać pozycje, które zaczynają się od podanego tekstu, należy skorzystać z funkcji `StartsWith` języka C#.

```
{Nazwa}.StartsWith("ABC")  
{Nazwa}.StartsWith("ABC") == true
```

Powyższe warunki są równoznaczne i będą spełnione, jeśli w kolumnie „Nazwa” znajdą się teksty zaczynające się od „ABC”, czyli np. „ABC s.c.” albo „ABCDE S.A.”. Wielkość liter ma tu znaczenie, a polskie znaki są wymagane.

Aby uzyskać pozycje, które nie zaczynają się od podanego tekstu, należy użyć takich samych wyrażeń jak powyżej, ale zanegować ich wynik.

```
!{Nazwa}.StartsWith("ABC")  
{Nazwa}.StartsWith("ABC") == false
```

W obu powyższych przypadkach pokazane zostaną tylko pozycje, w których tekst w kolumnie „Nazwa” nie zaczyna się od „ABC”.

Kolumny zawierające daty

Daty można porównywać za pomocą operatorów arytmetycznych: `=`, `!=`, `>`, `<`, `>=`, `<=`. Należy tu pamiętać, że to, że jedna data jest "większa" od drugiej, oznacza, że jest późniejsza.

Podczas porównywania dat ze stałą należy wprowadzać daty w formacie `dd-MM-yyyy` albo `yyyy-MM-dd`, gdzie `dd` to numer dnia zapisany za pomocą dwóch cyfr, `MM` to numer miesiąca zapisany za pomocą dwóch cyfr, a `yyyy` to numer roku zapisany za pomocą czterech cyfr. Data powinna być zapisana bez cudzysłowu, jak w przykładzie poniżej.

```
{Data wystawienia} > 01-11-2020
```

Taki warunek będzie spełniony, jeśli data w kolumnie „Data wystawienia” będzie większa, czyli późniejsza niż 1 listopada 2020. Na liście będą zatem widoczne dokumenty wystawione 2 listopada, ale nie te, które wystawiono 1 października.

Aby odwołać się w wyrażeniu do bieżącej daty, należy skorzystać z komponentów języka C#, a mianowicie własności `DateTime.Today`, pod którą ukrywa się bieżąca data, lub `DateTime.Now`, w której znajduje się bieżąca data wraz z aktualną godziną. Poniżej znajdują się przykłady wyrażeń zbudowanych w oparciu o `DateTime.Today` i `DateTime.Now`.

Pozycje dzisiejsze, czyli takie, w których „Data wystawienia” jest dzisiejsza:

```
{Data wystawienia} = DateTime.Today
```

Pozycje najnowsze, czyli takie, w których „Data wystawienia” jest późniejsza niż „cztery godziny temu” (to wyrażenie zadziała tylko pod warunkiem, że „Data wystawienia” poza datą zawiera też dokładną godzinę wystawienia):

```
{Data wystawienia} > DateTime.Now.AddHours(-4)
```

Pozycje z ostatniego tygodnia, czyli takie, w których „Data wystawienia” jest późniejsza niż „siedem dni temu”:

```
{Data wystawienia} > DateTime.Now.AddDays(-7)
```

Na wartościach w kolumnach datowych można również operować tak, jak na zmiennych typu `DateTime` w języku C#. Można w ten sposób zbudować na przykład takie wyrażenie, które będzie spełnione przez pozycje, w których „Data wystawienia” to 10. dzień miesiąca:

```
{Data wystawienia}.Day = 10
```

Więcej operacji i szczegółowe informacje można znaleźć w dokumentacji typu DateTime:

<https://docs.microsoft.com/en-us/dotnet/api/system.datetime>

Kolumny zawierające liczby

W kolumnach liczbowych na wartościach kolumn działa się za pomocą operatorów arytmetycznych: =, !=, >, <, >=, <=, %. Stałe do porównania należy wprowadzać bez cudzysłowu i bez spacji oddzielających grupy liczb (czyli „1000”, a nie „1 000”). Separatorem dziesiętnym może być zarówno przecinek, jak i kropka. Poniżej przedstawiono przykłady.

Pozycje, w których wartość w kolumnie „Brutto” jest większa od 15000:

```
{Brutto} > 15000
```

Pozycje, w których wartość w kolumnie „Postęp prac” jest większa od 50%:

```
{Postęp prac} > 50
```

Na wartościach w kolumnach datowych można również operować tak, jak na liczbach stało- lub zmiennoprzecinkowych w języku C#.

Pozycje, w których wartość w kolumnie „Precyzja waluty” jest parzysta:

```
{Precyzja waluty} % 2 = 0
```

Pozycje, w których wartość w kolumnie „Brutto” powiększona o 100 jest mniejsza niż 700:

```
{Brutto} + 100 > 700
```

Szczegółowe informacje można znaleźć w dokumentacji języka C#, w częściach poświęconych typom liczbowym i operatorom:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/integral-numeric-types>

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/floating-point-numeric-types>

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators>

Kolumny zawierające wartości logiczne („TAK”/„NIE”, „T”/„N”, „PRAWDA”/„FAŁSZ”)

W kolumnach logicznych zazwyczaj mamy do czynienia z dwiema możliwymi wartościami: prawdą i fałszem, które prezentowane są za pomocą tekstów typu „TAK”/„NIE”, „T”/„N”, „PRAWDA”/„FAŁSZ” albo ikonek (np. ✓ dla prawdy, brak ikonki dla fałszu). Niezależnie od tego, jaki jest sposób prezentacji wartości w kolumnie, operacje na nich wykonuje się zawsze tak samo. Główną operacją jest sprawdzenie, czy wartość w kolumnie jest równa „prawdzie”. Można zrobić to na dwa sposoby:

```
{Aktywna}
```

```
{Aktywna} = true
```

Powyższe warunki są równoznaczne i zostaną spełnione przez pozycje, w których w kolumnie „Aktywna” będzie wartość równa „prawdzie”, niezależnie od tego, czy będzie to zaprezentowane tekstem „TAK” lub innym czy też jakąś ikonką.

Aby sprawdzić, czy wartość w kolumnie jest fałszem, należy posłużyć się jednym z poniższych sposobów:

```
!{Aktywna}  
{Aktywna} = false
```

W przypadku, w którym kolumna logiczna może mieć trzy wartości, czyli prawdę, fałsz i null (brak wartości), wyrażenia muszą wyglądać następująco:

```
{Kolumna trzystanowa} = true  
{Kolumna trzystanowa} = false  
{Kolumna trzystanowa} = null
```

Użycie zapisu bez znaku = (np. `!{Kolumna trzystanowa}`) będzie w tym wypadku błędem.

Łączenie warunków

Warunki w wyrażeniu można łączyć przy pomocy operatorów AND, czyli „ORAZ”, i OR, czyli „LUB”. Operator AND można także zapisać jako &&, a OR jako |. Warunki są sprawdzane po kolei od lewej do prawej, przy czym AND ma pierwszeństwo przed OR oraz można używać nawiasów. W poniższym przykładzie użycie nawiasów sprawia, że zwracane są istotnie różne wyniki.

```
{Nazwa}.Contains("Pomadka") OR {Nazwa}.Contains("perfum") AND {Stan} > 500  
  
({Nazwa}.Contains("Pomadka") OR {Nazwa}.Contains("perfum")) AND {Stan} > 500
```

W pierwszym przypadku warunki połączone operatorem AND są rozpatrywane w pierwszej kolejności, więc całość oznacza wybór między pomadkami albo perfumami o stanie powyżej 500. Pomadki o stanie poniżej 500 pojawią się zatem na liście wyników. W drugim przypadku wybieramy pomadki lub perfumy, a dopiero potem sprawdzamy, czy stan jest większy od 500.