**Wikiprint Book**

**Title: [wiki:UserGuide Przewodnik ...**

**Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - UserGuide/AdvancedConfiguration/DefiningReports/ReportParams**

**Version: 34**

**Date: 09/05/25 11:57:10**

## Table of Contents

*Przewodnik użytkownika/Tworzenie raportów w SQL? > Filtry dla raportów* 3
- *Menu* 3
- *Wstęp* 3
- *Standardowe parametry* 3
- *Obsługiwane tokeny* 4
- *Definiowanie filtrów* 6
- *Ważne uwagi* 7
- *Definicja nowego filtra* 7
- *Typ opcji* 9
- *Jeden filtr wiele raportów* 12
- *Dynamiczne grupowanie* 12
- *Przydatne kwerendy SQL do filtrów* 13
  - *Drzewko miesięcy* 13
    - *Data wystawienia faktury* 13
    - *Filtr okresu urlopu* 13
  - *Filtr drzewka* 13
    - *MPK na fakturach* 13
    - *RK na fakturach* 13
  - *Inne* 14

# Przewodnik użytkownika/Tworzenie raportów w SQL? > Filtry dla raportów

*Informacje o wprowadzenie funkcjonalności:*

| Wersja systemu | Wersja modułu/funkcji | Data kompilacji | Zmiany | Opis |
|---|---|---|---|---|
| **4.7.7** | 0.5 | 19.03.2015 | Zmiana | Możliwość publikacji w wielu raportach |

**Menu**


1. Wstęp
2. Standardowe Parametry
3. Obsługiwane tokeny
4. Definiowanie filtrów
5. Wazne uwagi
6. Definicja nowego filtra
7. Typ opcji
8. Jeden filtr wiele raportów
9. Dynamiczne grupowanie
10. Przydatne kwerendy SQL do filtrów a) Drzewko miesięcy b) Filtr drzewka


## Wstęp

Moduł raportów w systemie eDokumenty umożliwia filtrowanie danego raportu poprzez określenie parametrów na dwa sposoby:

- standardowe parametry
- utworzenie filtrów przez użytkownika

## Standardowe parametry

Standardowe parametry to specjalne tokeny dzięki, którym możemy obsłużyć przekazywane dane z listy, kartoteki lub poprzez formularz parametrów do raportu.

Należy zwrócić uwagę na miejsce z którego jest wywoływany raport gdyż ma to wpływ na definicję operatora SQL przed takim tokenem. Warunki stosowanie operatorów:

| Operator | Kiedy stosować | Przykład parametru |
|---|---|---|
| **IN** | dla listy gdyż możemy otrzymać kilka zaznaczonych elementów.<br>Token zostanie zamieniony na identyfikatory z danej listy np.: 1,2,3,4 | **Miejsce:** Lista klientów<br>**Parametr:** contid IN ({CONTID}) |

| | | |
|---|---|---|
| **znak =** | dla kartoteki przekazywana jest tylko jedna wartość do parametru | **Miejsce:** Kartoteka klienta<br>**Parametr:** contid = {CONTID} |

W przypadku jeśli w definicji raportu użyjemy np.: {CONTID} a raport zostanie wywołany z modułu (przekazane zostaną zaznaczone elementy) wtedy token zostanie zamieniony na pierwszy z zaznaczonych elementów.

W przypadku niewłaściwego podpięcia raportu lub niewłaściwego użycia operatora raport wygeneruje błąd.

### Obsługiwane tokeny

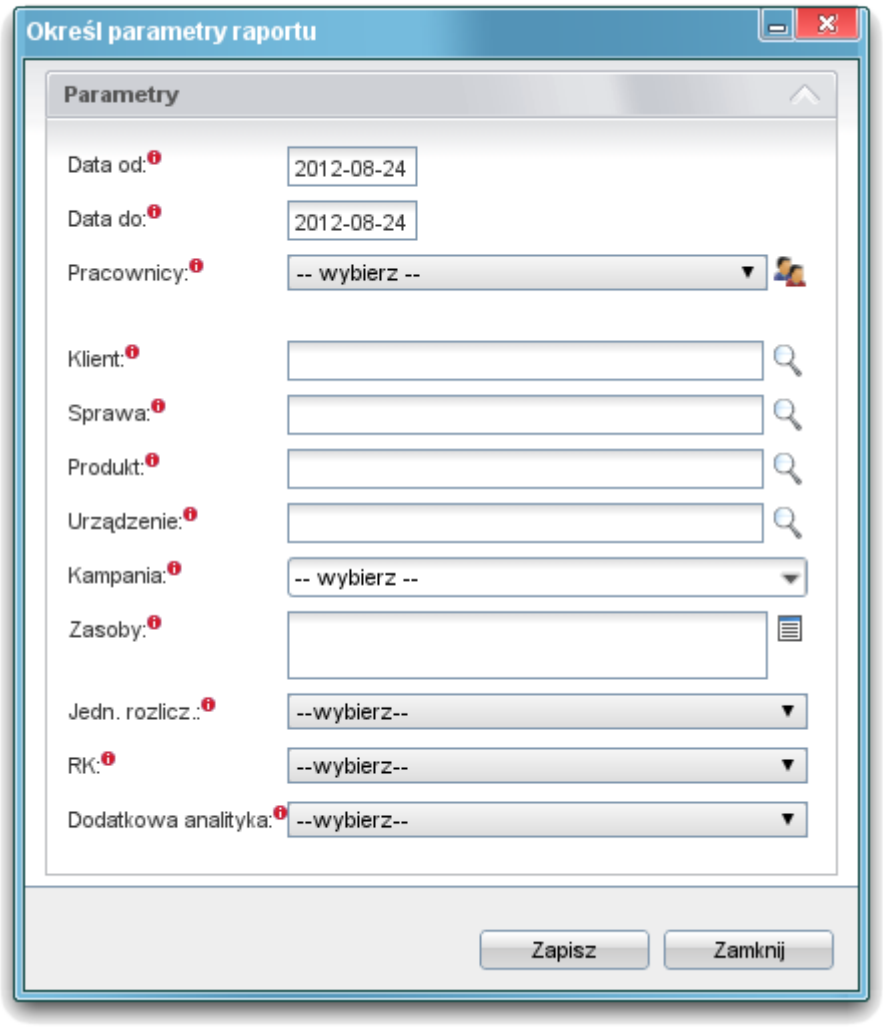
Obecnie obsługiwane tokeny to:

| Nazwa | Opis | Wartość domyślna | Określanie na formularzu |
|---|---|---|---|
| **{DATE_FROM}** | Data od | Początek bieżącego miesiąca | TAK |
| **{DATE_TO}** | Data do | Koniec bieżącego miesiąca | TAK |
| **{USR_ID}** | Pracownik | Pracownik z listy | TAK |
| **{LOGGED_USR_ID}** | Zalogowany pracownik | Id zalogowanego pracownika (users.usr_id) | NIE |
| **{ENT_ID}** | Identyfikator podmiotu | Identyfikator podmiotu | NIE |
| **{FILTER_STRING}** | Ciąg filtrujący | 'TRUE' | TAK - definicja filtrów to określa |
| **{ACORID}** | Jednostka rozliczeniowa | 0 | TAK |
| **{TOVCID}** | Miejsce powstawania kosztów | 0 | TAK |
| **{ADANID}** | Dodatkowa analityka | 0 | TAK |
| **{CONTID}** | Klient | Pierwszy nie usunięty klient z listy (sortowany po contid) | TAK |

| | | | |
|---|---|---|---|
| **{CONTIDS}** | Klient | Pierwszy nie usunięty klient z listy (sortowany po contid) | TAK |
| **{DOC_ID}** | Dokument (z listy lub kartoteka) | 0 | NIE |
| **{DOC_IDS}** | Dokument (z listy lub kartoteka) | 0 | NIE |
| **{DOCIDS}** | Dokument (z listy lub kartoteka) | 0 | NIE |
| **{PRC_ID}** | Sprawa | 0 | TAK |
| **{PRC_IDS}** | Sprawa | 0 | TAK |
| **{PRCIDS}** | Sprawa | 0 | TAK |
| **{EVNTID}** | Zdarzenie (z listy lub kartoteka) | 0 | NIE |
| **{EVNTIDS}** | Zdarzenie (z listy lub kartoteka) | 0 | NIE |
| **{CAMPID}** | Kampania (z listy lub kartoteka) | 0 | TAK |
| **{CAMPIDS}** | Kampania (z listy lub kartoteka) | 0 | TAK |
| **{DEVCID}** | Urządzenie (z listy lub kartoteka) | 0 | TAK |
| **{DEVCIDS}** | Urządzenie (z listy lub kartoteka) | 0 | TAK |
| **{DEPOID}** | Produkt (z listy lub kartoteka) | 0 | TAK |

| | | | |
|---|---|---|---|
| **{DEPOIDS}** | Produkt (z listy lub kartoteka) | 0 | TAK |
| **{RES_ID}** | Zasób (z listy lub kartoteka) | 0 | TAK |

W momencie jeśli w danym raporcie zostanie użyty taki token to system wygeneruje odpowiednie pole na formularzu parametrów do raportu - jeśli jest obsługiwane. Poniższy zrzut ekranu przedstawia wszystkie tokeny możliwe do określania poprzez formularz.




(*Parametry*)

### Definiowanie filtrów

Od wersji 4.0 system eDokumenty umożliwia definiowanie własnych parametrów do raportów określanych jako Filtry.

Na wstępie należy wyszukać interesujący nas raport lub dodać nowy Tworzenie raportów w SQL?.

Przykładowa definicja raportu (kod SQL):

```
SELECT
   contid,
   name_1,
   adddat,
   macrtk,
```

```
   is_del
FROM
   contacts
WHERE
   ent_id = {ENT_ID}
   AND {FILTER_STRING}
```

### Ważne uwagi

Aby filtrowanie było możliwe wymagane jest dodanie specjalnego tokenu **{FILTER_STRING}**, który w momencie akceptacji filtrów zostanie zamieniony na warunki SQL zgodne z definicją filtrów. Jeśli w definicji raportu zabraknie tego tokenu to pomimo definicji filtrów nie będą one dostępne na panelu parametrów.

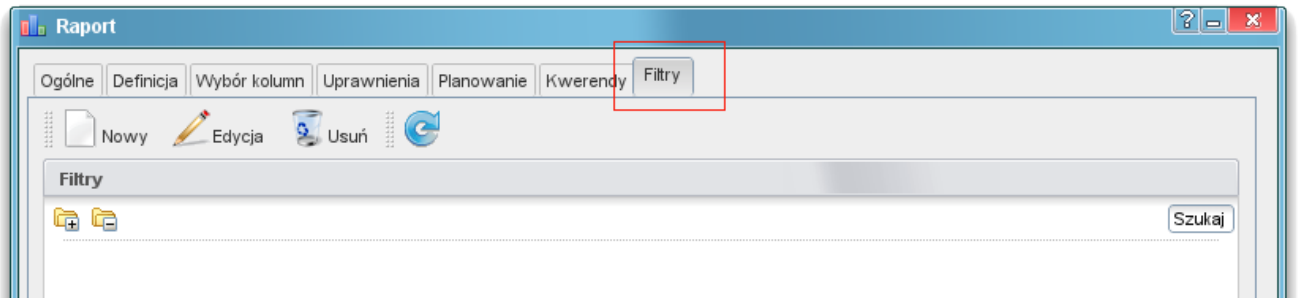
Istotną kwestią jest również miejsce w którym jest dodany token {FILTER_STRING} oraz operator jaki go poprzedza (AND/OR zalecany AND). Być może przykład będzie dość abstrakcyjny jednak proszę zwrócić uwagę na zapytanie zagnieżdżone:

```
SELECT *
FROM
   (SELECT
       contid,
       name_1,
       adddat,
       macrtk,
       is_del
   FROM
       contacts
   WHERE
       ent_id = {ENT_ID} LIMIT 100) AS foo
   AND {FILTER_STRING}
```

Ważne jest aby przy takiej definicji raportu filtry wskazywały na kolumny umieszczone w głównym zapytaniu.

### Definicja nowego filtra

Po otwarciu dialoga raportu powinna nam się pojawić dodatkowa zakładka - "Filtry".



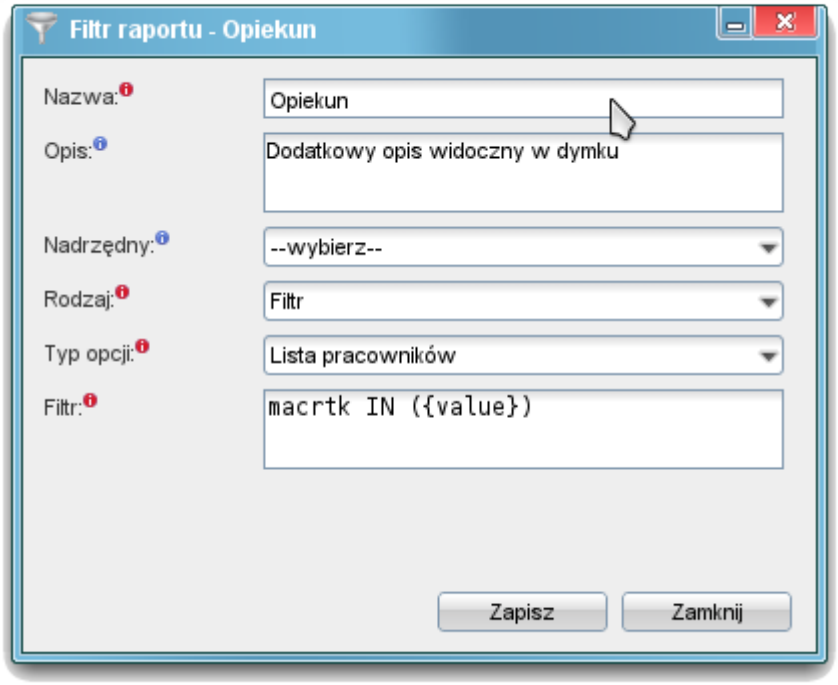

(*Panel "Filtry"*)

Przyciski na widocznym zrzucie umożliwiają standardowe zarządzanie filtrami:

| Nazwa przycisku | Opis |
|---|---|
| **Nowy** | definicja nowego filtra/grupy |

| | |
|---|---|
| **Edycja** | edycja istniejącego filtra/grupy możliwa również poprzez dwukrotne kliknięcie elementu drzewka |
| **Usuń** | sunięcie filtra lub grupy (usuwane są też wszystkie podrzędne) |
| **Odśwież listę** | odświeża listę filtrów |

Klikając przycisk "Nowy" otrzymujemy formularz dzięki, której możemy zdefiniować nowy filtr lub grupę (zależne od wyboru pola Rodzaj - poniższy zrzut ekranu).




(*Definicja filtra*)

Opis pól:

| Nazwa pola | Opis |
|---|---|
| **Nazwa** | nazwa filtra lub grupy filtrów w zależności co zostanie wybrane w polu rodzaj |
| **Opis** | dodatkowa informacja wyświetlania po najechaniu na nazwę filtra w panelu filtrów raportu (dostępny na wygenerowanym raporcie) |
| **Nadrzędny** | w przypadku jeśli dany filtr lub grupa ma należeć do grupy należy wybrać z dostępnej listy |

| | |
|---|---|
| **Rodzaj** | **Grupa** - grupuje kilka filtrów lub innych grup<br>**Filtr** - konkretny filtr - filtry mogą być definiowane bez elementu nadrzędnego lub mogą należeć do grupy. Filtr nie może należeć do innego filtra |
| **Typ opcji** | rodzaj wygenerowanego pola na panelu filtrów w wygenerowanym raporcie. Szczegóły poniżej |
| **Filtr** | wyrażenie filtrujące. W zależności od typ opcji może wymagać podanie tokenu {value} - szczegóły poniżej |
| **Wartości dla listy** | tylko dla typu: Lista jednokrotnego wyboru, Lista wielokrotnego wyboru, Pola radiowe - kwerenda SQL, gdzie pierwsza wartość to klucz, który **musi być typu INTEGER** natomiast druga to opis wartości. Można stosować następujące tokeny:<br>{LOGGED_USER} - identyfikator zalogowanego pracownika (users.usr_id)<br>{LOGGED_ORUNID} - identyfikator jednostki zalogowanego pracownika (organization_units.orunid)<br>{ENT_ID} - identyfikator podmiotu (podmioty.id_pod)<br>{CURRENT_DATE} - bieżąca data w formacie YYYY-MM-DD<br>{CURRENT_TIME} - bieżący czas w formacie HH:MM |

### Typ opcji

Poniżej znajduje się aktualna lista typów opcji oraz przykład definicji:

| Typ opcji | Opis | Przykład filtru | Wymaga tokenu {value} |
|---|---|---|---|
| **Wartość BOOLEAN** | Zastosowanie dla kolumn typu boolean gdzie wartość to TRUE lub FALSE | is_del IS {value} | Tak |
| **Wartość TEXT** | Dowolne pole tekstowe do którego chcemy zastosować wyszukiwanie | name_1 ~* E'{value}' | Tak |
| **Pole zaznaczane** | Jedna opcja zaznaczenia filtruje tylko wtedy gdy jest zaznaczony. Jeśli podamy w filtrze token {value} zostanie zamieniony na TRUE | is_del IS TRUE to samo da nam is_del IS {value} ale is_del IS FALSE nie jest tym samym co is_del IS {value} - {value} jest zawsze zamieniane na TRUE | Nie ale może być patrz przykład |

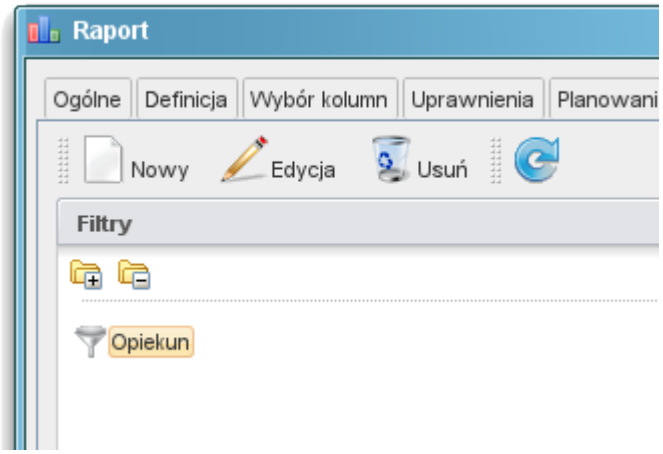
| | | | |
|---|---|---|---|
| **Lista jednokrotnego wyboru** | Pole typu Select gdzie można wybrać tylko jedną wartość. Wymagane jest zadeklarowanie SQL dla listy opcji | adduid = {value} | Tak |
| **Lista wielokrotnego wyboru** | Lista z opcjami do wyboru | adduid | Nie tylko należy zadeklarować kolumnę której filtr dotyczy |
| **Pola radiowe** | Lista z opcjami do wyboru | adduid | Nie tylko należy zadeklarować kolumnę której filtr dotyczy |
| **Pole daty z godziną** | Wizualizacja tego filtru posiada dodatkowe pole do określania godziny jeśli jest wymagana taka do dokładność | adddat >= '{value}' lub można adddat >= '2012-01-01 12:00' | Raczej tak niepodanie tokenu {value} jest niecelowe |
| **Pole daty bez godziny** | Brak pola godzina | adddat >= '{value}' lub można adddat >= '2012-01-01' | Raczej tak niepodanie tokenu {value} jest niecelowe |
| **Wybór miesiąca oraz roku** | Umożliwia wybranie roku oraz miesiąca. Zwracana wartość to rok plus miesiąc oraz 1 dzień czyli np. 2012-02-01. Umożliwia to budowanie zakresów miesięcy jeśli zdefiniujemy drugi filtr tego typu w grupie | adddat >= '{value}' | Tak |
| **Kontakt z książki adresowej** | Zwraca identyfikator (contacts.contid) klienta | contid = {value} | Tak |
| **Lista spraw** | Zwraca identyfikator sprawy (processes.prc_id) | prc_id = {value} | Tak |
| **Osoba kontaktowa** | Zwraca identyfikator osoby kontaktowej (contact_persons.copeid) | copeid = {value} | Tak |

| | | | |
|---|---|---|---|
| **Urządzenie** | Zwraca identyfikator urządzenia (devices.devcid) | devcid = {value} | Tak |
| **Projekt** | Zwraca identyfikator projektu (projects.projid) | projid = {value} | Tak |
| **Lista pracowników** | Zwraca identyfikator wybranych pracowników (users.usr_id). **WAŻNE - wymagane jest aby był operator IN w wyrażeniu filtrującym gdyż wartości są zwracane w postaci 1,2,3,4** | adduid IN ({value}) | Tak |
| **Drzewko** | Zwraca identyfikator wybranego elementu drzewka. Napełniane kwerendą podaną poniżej w przykładach | category = {value} | Tak |

W moim przypadku chcę filtrować listę klientów według parametru opiekun dlatego:

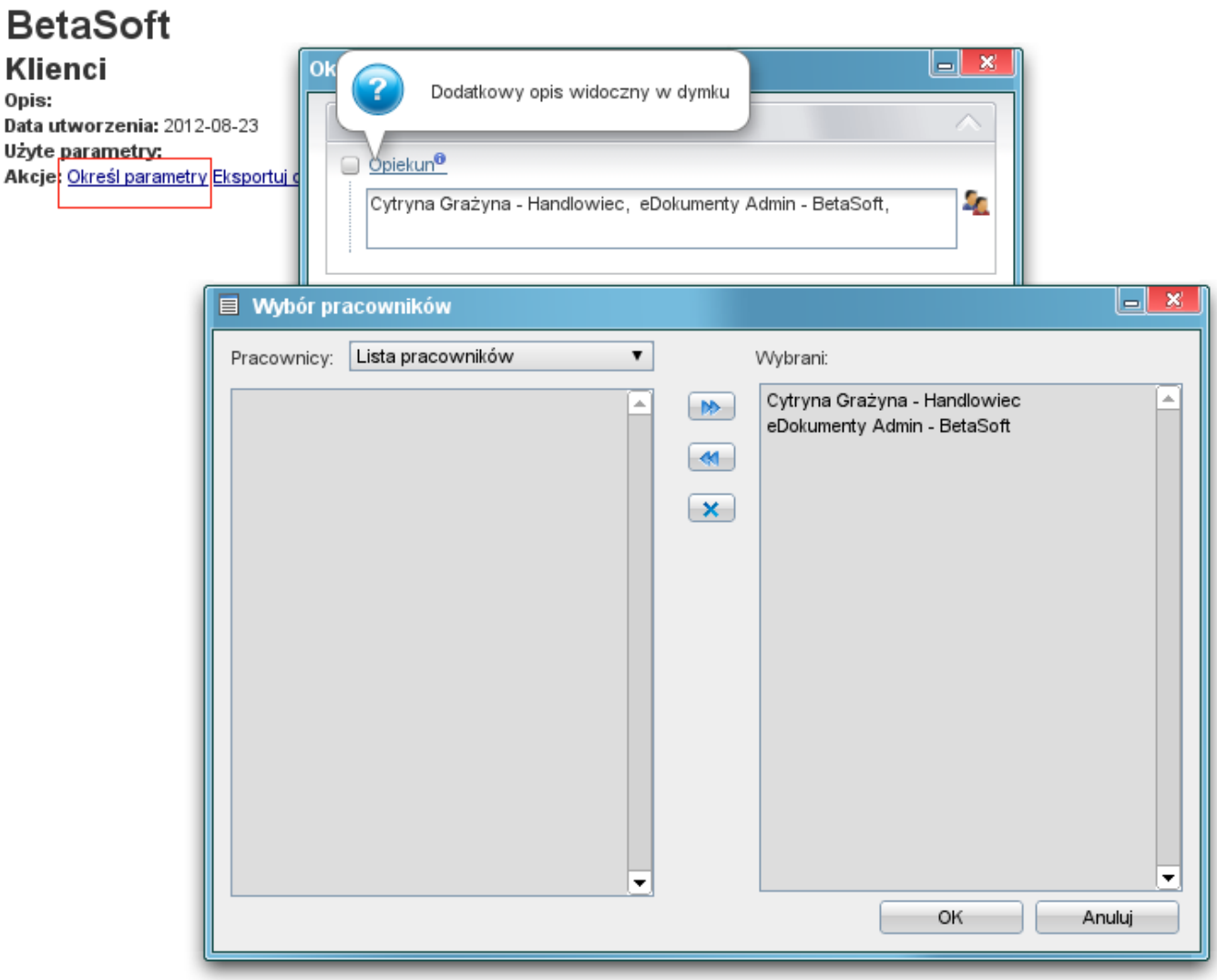
- Typ opcji : Lista pracowników
- Filtr: macrtk IN ({value}) - pole macrtk przechowuje id opiekuna (users.usr_id)

Po zapisaniu filtr jest dostępny z poziomu listy filtrów danego raportu:




(*Nowy filtr na liście filtrów w raporcie*)

Oraz dodatkowo po wygenerowaniu raportu klikając przycisk Określ parametry otrzymujemy formularz jak na zrzucie poniżej:




(*Filtr na wygenerowanym raporcie*)

Po zatwierdzeniu (przycisk Zapisz na formularzu Określ parametru raportu) raport zostanie ponownie wygenerowany uwzględniając nasz wybór.

### Jeden filtr wiele raportów

Nowa wersja 4.7.7 umożliwia publikację filtrów w wielu raportach. Dodatkowo został dodany słownik definicji filtrów dostępny z poziomu modułu Raporty -> Akcje -> Edycja definicji filtrów lub z poziomu zakładki Filtry w raporcie na pasku zadań.

Przy dodawaniu filtru bezpośrednio z raportu system tworzy definicję filtru oraz publikuje go w danym raporcie w kontekście, którego aktualnie pracujemy. Zachowanie to możemy zmienić poprzez dodawanie już istniejącej definicji filtru do danego raportu. Można to zrobić poprzez opcję w menu Nowy na zakładce Filtry "Dodaj istniejący (powiązanie)". Z otwartego słownika wybieramy filtry lub grupę i klikamy Dodaj lub Dodaj i zamknij. Ważne jest, że jeśli publikujemy (dodajemy) element, który ma elementy podrzędne nastąpi automatyczne dodanie również tych elementów podrzędnych.

Publikować (dodawać powiązania) można również poprzez metodę drag & drop danego elementu drzewka definicji filtrów na panel drzewka filtrów w raporcie.

### Dynamiczne grupowanie

Aby na panelu filtrów widoczna była opcja dynamicznego grupowania (po kolumnach zdefiniowanych na panelu Wybór kolumn) w definicji SQL raportu należy dodać token {GROUP_BY}. Jest to jednak token sterujący dlatego należy go poprzedzić --. Pozycja tokena {GROUP_BY} nie ma znaczenia ważne aby był zakomentowany.

Przykład SQL

```
SELECT * FROM contacts --{GROUP_BY}
```

## Przydatne kwerendy SQL do filtrów

### Drzewko miesięcy

#### Data wystawienia faktury

```
crtdat {value}

select coalesce(
(select min(coalesce (d.adddat) -interval '1 day') ::date
FROM
documents d
where d.dctpid in (2) and d.is_del is false and d.gostof is null), current_date::date-interval '1 day')
union
select coalesce(
(select max(d.adddat )::date
FROM
documents d
where d.dctpid in (2) and d.is_del is false and d.gostof is null),current_date::date)
```

#### Filtr okresu urlopu

```
pocz_urlop in (select s FROM generate_series('2008-03-01 00:00'::timestamp,'2050-03-04 00:00', '1 day') as S where s {valu
or
kon_urlop in (select s FROM generate_series('2008-03-01 00:00'::timestamp,'2050-03-04 00:00', '1 day') as S where s {value
or
(pocz_urlop < (select min(s) FROM generate_series('2008-03-01 00:00'::timestamp,'2050-03-04 00:00', '1 day') as S where s
and
kon_urlop > (select max(s) FROM generate_series('2008-03-01 00:00'::timestamp,'2050-03-04 00:00', '1 day') as S where s {v

select coalesce(
(select min(CAST(f1.data__ AS date)  -interval '1 day') ::date
FROM documents doc
LEFT JOIN features_text_view f1 ON doc.doc_id = f1.tbl_id AND f1.featid = 4 /*data od*/
where dctpid = 24 AND gostof IS NULL and  copyof is null AND  doc.is_del IS FALSE and f1.data__ is not null), current_date
union
select coalesce(
(select max(CAST(f1.data__ AS date))::date
FROM documents doc
LEFT JOIN features_text_view f1 ON doc.doc_id = f1.tbl_id AND f1.featid = 4 /*data od*/
where dctpid = 24 AND gostof IS NULL and  copyof is null AND  doc.is_del IS FALSE and f1.data__ is not null),current_date:
```

### Filtr drzewka

#### MPK na fakturach

Filtr pozwala na znajdowanie faktur kosztowych ze względu na wybrane MPK

```
keyval in (select doc_id from vatnote_costs where povcid in ({value}))

select povcid as keyval, prn_id,  place_||' '||coalesce (dscrpt,'') name__, 'FOLDER' AS icon__ FROM places_of_vcosts where
EXTRACT(year FROM current_date)  and is_del is false
order by name__
```

#### RK na fakturach

Filtr pozwala na znajdowanie faktur kosztowych ze względu na wybrane RK

```
keyval in (select doc_id from vatnote_costs where tovcid in ({value}))

select tovcid as keyval, prn_id,  type__||' '||coalesce (dscrpt,'') name__, 'FOLDER' AS icon__ FROM types_of_vcosts where
EXTRACT(year FROM current_date)  and is_del is false
order by name__
```

**Inne**

MAGAZYNY

```
SELECT wahaid, name__ FROM warehouses WHERE is_del IS false ORDER BY prior_
```

GRUPY PRODUKTÓW

```
SELECT pgr_id, name__ FROM product_groups
WHERE is_act IS TRUE AND is_del IS FALSE AND pgr_id NOT IN (SELECT prn_id FROM product_groups) ORDER BY prn_id, name__
```

KOMÓRKI ORGANIZACYJNE

```
-- z ograniczeniem do tych do których user ma prawo
SELECT orunid, orunsm AS caption
FROM organization_units
WHERE is_del IS FALSE
AND ndetpe = 'ORGCELL'
AND chkUsrOrgAcc({LOGGED_USR_ID}, orunid)
ORDER BY orunsm

-- Wybór komórek typu Komórka organizacyjna (same działy)
SELECT orunid, COALESCE(orunsm, '') || ' - ' || ndenam AS stanowisko
FROM orgtree_view
WHERE orunid > 0 AND ndetpe = 'ORGCELL'
AND orunsm IS NOT NULL AND is_del IS FALSE
```

OPIEKUN SPRAWY (lista pracowników)

```
Filtr = rspuid={value}
Do listy: SELECT usr_id, fullnm FROM orgtree_view WHERE is_del is FALSE
```

STATUSY (lista)

```
-- SPRAWY
Filtr = state_ in ({value})
SELECT tpstid, dscrpt FROM types_of_processes_states WHERE clsnam LIKE 'PROCESS%' ORDER BY state_

-- DOKUMENTY typu dctpid = 2 (Faktury)
SELECT tpstid, dscrpt FROM types_of_processes_states
WHERE clsnam = 'DOCUMENT|2' ORDER BY state_
```

TERMIN PLANOWANY > OD (data)

```
foreda >= '{value}'::date
```

WARTOŚĆ SPRAWY (tekstowy)

```
forepa > COALESCE(NULLIF('{value}', '')::int,0)
```

```
-- Osoby do których użytkownik ma prawo
SELECT ov.usr_id, ov.usrnam FROM orgtree_view ov where
```

```
chkUsrOrgAcc({LOGGED_USR_ID}, ov.orunid) IS TRUE


-- Wartości cechy osób do których uzytkownik ma prawo
SELECT ftopid, ftopnm
FROM
features_options
INNER JOIN features_options_def USING(fodfid)
LEFT JOIN users u ON u.initls = ftopnm
LEFT JOIN orgtree_view ov ON ov.usr_id = u.usr_id
WHERE featid = 91
AND (CASE WHEN (SELECT count(*) > 0 FROM users_link_group
WHERE grp_id = 29 AND usr_id = {LOGGED_USR_ID}) THEN 1 = 1
ELSE chkUsrOrgAcc({LOGGED_USR_ID}, ov.orunid) IS TRUE END)
ORDER BY ftopnm

-- Wartości cechy tekstowej o featid 83
SELECT DISTINCT data__ AS value, data__ AS caption
FROM features_text_view
WHERE featid = 83
ORDER BY data__

-- Wartości do listy wyboru z cechy - lista opcji
SELECT ftopid, ftopnm
FROM
features_options
INNER JOIN features_options_def USING(fodfid)
WHERE featid = 66
ORDER BY ftopnm

+ przykład wartości w polu Filter:
keyval IN (SELECT tbl_id FROM features_opt_view fov WHERE fov.featid = 66 AND fov.ftopid IN ({value}))

-- Format zapytania do zwrócenia drzewka
-- Może być wykonane na dowolnej strukturze danych, ważne by zwracało elementy: KLUCZ, RODZIC, NAZWA, TYP.
SELECT c.strcid AS keyval, c.prn_id, c.strnam AS name__,
CASE WHEN (SELECT count(*) FROM cregisters.creg_structure_elements c2
WHERE c2.prn_id = c.strcid) > 0 THEN 'FOLDER' ELSE 'ITEM' END AS icon__
FROM cregisters.creg_structure_elements c WHERE c.is_del IS FALSE


-- Filtr grupy produktów dla CRM
-- filtry typu lista wielokrotnego wyboru
SELECT prc_id FROM processes WHERE prc_id IN (SELECT prc_id FROM processes
INNER JOIN fk_elements fe USING(prc_id)
INNER JOIN depository dep USING (depoid)
WHERE fe.is_del IS FALSE
AND dep.pgr_id IN (13))

--wartości dla listy
SELECT pgr_id, name__ FROM product_groups WHERE is_act IS TRUE AND is_del IS FALSE ORDER BY name__
```