

Wikiprint Book

Title: [wiki:UserGuide Przewodnik ...

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

Version: 34

Date: 06/05/26 22:14:32

## Table of Contents

<i>Przewodnik użytkownika/Tworzenie raportów w SQL? &gt; Filtry dla raportów</i>	3
<i>Menu</i>	3
<i>Wstęp</i>	3
<i>Standardowe parametry</i>	3
<i>Obsługiwane tokeny</i>	4
<i>Definiowanie filtrów</i>	7
<i>Ważne uwagi</i>	7
<i>Definicja nowego filtra</i>	7
<i>Typ opcji</i>	9
<i>Jeden filtr wiele raportów</i>	12
<i>Dynamiczne grupowanie</i>	12
<i>Przydatne kwerendy SQL do filtrów</i>	13
<i>Lista wielokrotnego wyboru</i>	13
<i>Filtr statusów faktur</i>	13
<i>Filtr działów</i>	13
<i>Drzewko miesięcy</i>	13
<i>Data wystawienia faktury</i>	13
<i>Filtr okresu urlopu</i>	13
<i>Filtr drzewka</i>	14
<i>MPK na fakturach</i>	14
<i>RK na fakturach</i>	14
<i>Inne</i>	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. [Ważne uwagi](#)
6. [Definicja nowego filtra](#)
7. [Typ opcji](#)
8. [Jeden filtr wiele raportów](#)
9. [Dynamiczne grupowanie](#)
- [Przydatne kwerendy SQL do filtrów](#)
  - [Lista wielokrotnego wyboru](#)
    1. Filtr statusów faktur
    2. Filtr działów
  - [Drzewko miesięcy](#)
    1. Data wystawienia faktury
    2. Filtr okresu urlopu
  - [Filtr drzewka](#)
    1. MPK na fakturach
    2. RK na fakturach
11. [Inne](#)

### 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 obsługiwać 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. Token zostanie zamieniony na identyfikatory z danej listy np.: 1,2,3,4	<b>Miejsce:</b> Lista klientów <b>Parametr:</b> contid IN ({{CONTID}})
znak =	dla kartoteki przekazywana jest tylko jedna wartość do parametru	<b>Miejsce:</b> Kartoteka klienta <b>Parametr:</b> 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

<b>{ADANID}</b>	Dodatkowa analityka	0	TAK
<b>{CONTID}</b>	Klient	Pierwszy nie usunięty klient z listy (sortowany po contid)	TAK
<b>{CONTIDS}</b>	Klient	Pierwszy nie usunięty klient z listy (sortowany po contid)	TAK
<b>{DOC_ID}</b>	Dokument (z listy lub kartoteka)	0	NIE
<b>{DOC_IDS}</b>	Dokument (z listy lub kartoteka)	0	NIE
<b>{DOCIDS}</b>	Dokument (z listy lub kartoteka)	0	NIE
<b>{PRC_ID}</b>	Sprawa	0	TAK
<b>{PRC_IDS}</b>	Sprawa	0	TAK
<b>{PRCIDS}</b>	Sprawa	0	TAK
<b>{EVNTID}</b>	Zdarzenie (z listy lub kartoteka)	0	NIE
<b>{EVNTIDS}</b>	Zdarzenie (z listy lub kartoteka)	0	NIE
<b>{CAMPID}</b>	Kampania (z listy lub kartoteka)	0	TAK
<b>{CAMPIDS}</b>	Kampania (z listy lub kartoteka)	0	TAK
<b>{DEVCID}</b>	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. W tokenach można używać modyfikatorów powodujących escapowanie wartości:

:as - escapuje do PHP

:pges - escapuje do prawidłowej składni postgresqla

:acs - escapuje apostrof i backslash

Prawidłowe użycie, np: {FILTER\_STRING:pges}

(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.

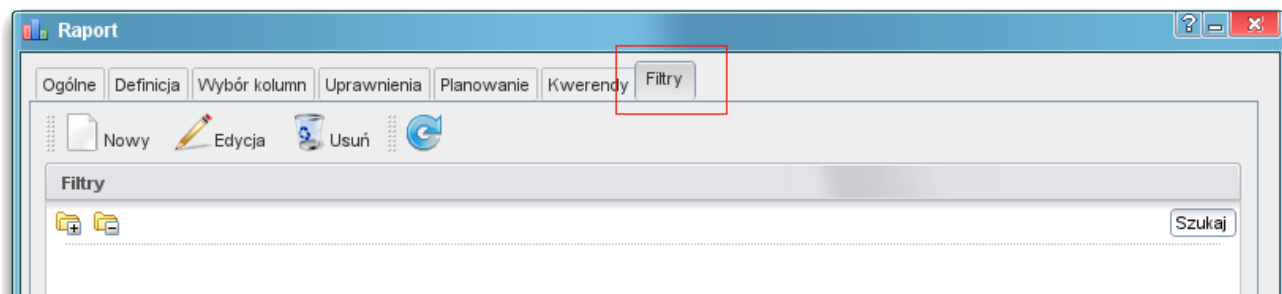
Istotną kwestią jest również miejsce w którym jest dodany token **{FILTER\_STRING}** oraz operator jaki go poprzedza (AND/OR zalecany AND). Być może przykład będzie dość abstrakcyjny jednak proszę zwrócić uwagę na zapytanie zagnieżdżone:

```
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
<b>Nowy</b>	definicja nowego filtra/grupy
<b>Edycja</b>	edycja istniejącego filtra/grupy możliwa również poprzez dwukrotne kliknięcie elementu drzewka
<b>Usuń</b>	sunięcie filtra lub grupy (usuwane są też wszystkie podrzędne)
<b>Odśwież listę</b>	odświeża listę filtrów

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

**Filtr raportu - Opiekun**

Nazwa:

Opis:

Nadrzędny:

Rodzaj:

Typ opcji:

Filtr:

Zapisz    Zamknij

(Definicja filtra)

Opis pól:

Nazwa pola	Opis
------------	------

<b>Nazwa</b>	nazwa filtra lub grupy filtrów w zależności co zostanie wybrane w polu rodzaj
<b>Opis</b>	dodatkowa informacja wyświetlania po najechnaniu na nazwę filtra w panelu filtrów raportu (dostępny na wygenerowanym raporcie)
<b>Nadrzędny</b>	w przypadku jeśli dany filtr lub grupa ma należeć do grupy należy wybrać z dostępnej listy
<b>Rodzaj</b>	<b>Grupa</b> - grupuje kilka filtrów lub innych grup <b>Filtr</b> - konkretny filtr - filtry mogą być definiowane bez elementu nadrzędnego lub mogą należeć do grupy. Filtr nie może należeć do innego filtra
<b>Typ opcji</b>	rodzaj wygenerowanego pola na panelu filtrów w wygenerowanym raporcie. Szczegóły poniżej
<b>Filtr</b>	wyrażenie filtrujące. W zależności od typ opcji może wymagać podanie tokenu {value} - szczegóły poniżej
<b>Wartości dla listy</b>	tylko dla typu: Lista jednokrotnego wyboru, Lista wielokrotnego wyboru, Pola radiowe - kwerenda SQL, gdzie pierwsza wartość to klucz, który <b>musi być typu INTEGER</b> natomiast druga to opis wartości. Można stosować następujące tokeny: {LOGGED_USER} - identyfikator zalogowanego pracownika (users.usr_id) {LOGGED_ORUNID} - identyfikator jednostki zalogowanego pracownika (organization_units.orunid) {ENT_ID} - identyfikator podmiotu (podmioty.id_pod) {CURRENT_DATE} - bieżąca data w formacie YYYY-MM-DD {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}
<b>Wartość BOOLEAN</b>	Zastosowanie dla kolumn typu boolean gdzie wartość to TRUE lub FALSE	is_del IS {value}	Tak

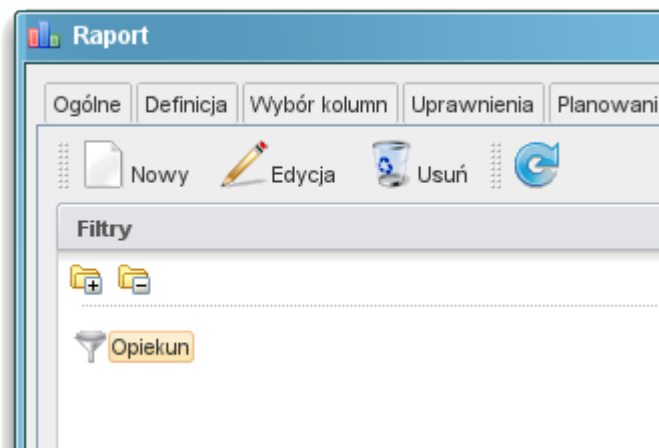
<b>Wartość TEXT</b>	Dowolne pole tekstowe do którego chcemy zastosować wyszukiwanie	name_1 ~* E'{value}'	Tak
<b>Pole zaznaczane</b>	Jedna opcja zaznaczenia filtruje tylko wtedy gdy jest zaznaczony. Jeśli podamy w filtrze token {value} zostanie zamieniony na TRUE	is_del IS TRUE to samo da nam is_del IS {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
<b>Lista jednokrotnego wyboru</b>	Pole typu Select gdzie można wybrać tylko jedną wartość. Wymagane jest zadeklarowanie SQL dla listy opcji	adduid = {value}	Tak
<b>Lista wielokrotnego wyboru</b>	Lista z opcjami do wyboru	adduid	Nie tylko należy zadeklarować kolumnę której filtr dotyczy
<b>Pola radiowe</b>	Lista z opcjami do wyboru	adduid	Nie tylko należy zadeklarować kolumnę której filtr dotyczy
<b>Pole daty z godziną</b>	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
<b>Pole daty bez godziny</b>	Brak pola godzina	adddat >= '{value}' lub można adddat >= '2012-01-01'	Raczej tak niepodanie tokenu {value} jest niecelowe
<b>Wybór miesiąca oraz roku</b>	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
<b>Kontakt z książki adresowej</b>	Zwraca identyfikator (contacts.contid) klienta	contid = {value}	Tak

<b>Lista spraw</b>	Zwraca identyfikator sprawy (processes.prc_id)	prc_id = {value}	Tak
<b>Osoba kontaktowa</b>	Zwraca identyfikator osoby kontaktowej (contact_persons.copeid)	copeid = {value}	Tak
<b>Urządzenie</b>	Zwraca identyfikator urządzenia (devices.devcid)	devcid = {value}	Tak
<b>Projekt</b>	Zwraca identyfikator projektu (projects.projid)	projid = {value}	Tak
<b>Lista pracowników</b>	Zwraca identyfikator wybranych pracowników (users.usr_id). <b>WAŻNE - wymagane jest aby był operator IN w wyrażeniu filtrującym gdyż wartości są zwracane w postaci 1,2,3,4</b>	adduid IN ({value})	Tak
<b>Drzewko</b>	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: macrck IN ({value}) - pole macrck 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:

## BetaSoft

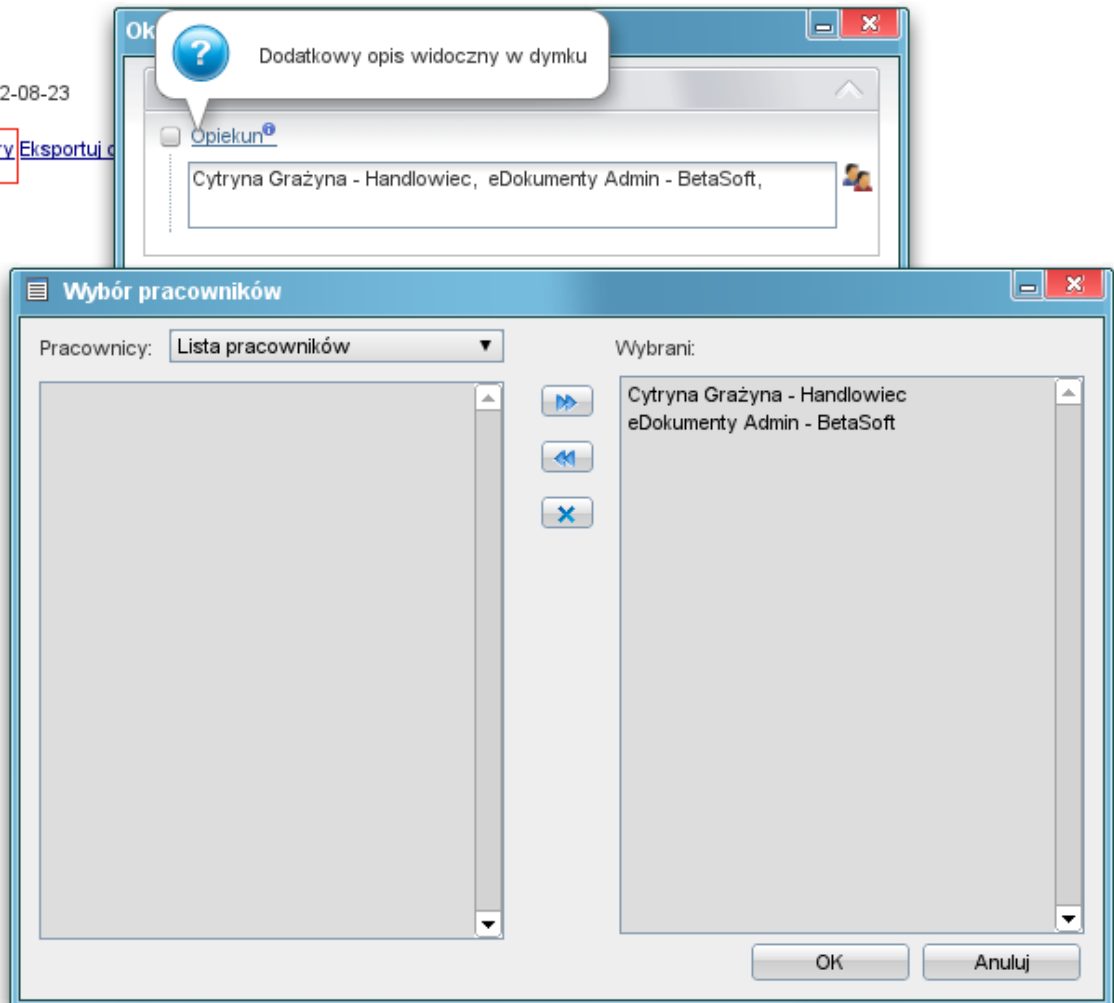
### Klienci

Opis:

Data utworzenia: 2012-08-23

Użyte parametry:

Akcje: [Określ parametry](#) [Eksportuj](#)



(Filtr na wygenerowanym raporcie)

Po zatwierdzeniu (przycisk Zapisz na formularzu Określ parametry 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

Raport ma pokazywać wyniki tylko jeśli wybrany jest jakiś filtr, dzięki temu przy dużej liczbie rekordów raport wyświetla brak wyników

```
case when ($${FILTER_STRING}$$ = 'TRUE' or $$${FILTER_STRING}$$ = '1=1') then keyval = 0 else {FILTER_STRING} end
```

### Lista wielokrotnego wyboru

#### Filtr statusów faktur

```
tpstid in ({value})

SELECT tpstid, dscrpt, * FROM types_of_processes_states
WHERE clsnam = 'DOCUMENT|2' ORDER BY state_
```

#### Filtr działań

Raport wymaga zdefiniowania dodatkowego pola

```
CASE WHEN u2.orunid IS NOT NULL AND u2.orunid > 0 THEN NULLIF(get_post(u2.orunid)::text, 'f'::text)::integer ELSE 0 END g
```

```
getpost::int IN ({value})

SELECT orunid, orunsm || ' - ' || ndenam AS caption
FROM organization_units
WHERE is_del IS FALSE
AND orunid > 0
AND ndetpe = 'ORGCELL'
AND (chkUsrOrgAcc({LOGGED_USR_ID}, orunid) OR {LOGGED_USR_ID} = 26)
ORDER BY orunsm
```

### 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 {value}
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 {value}
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 {value}

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.feaid = 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.feaid = 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 wahaaid, 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__

```