

Przykłady workflow

W tym miejscu publikowane będą przykładowe kwerendy użyteczne przy budowaniu warunków, przekazywaniu parametrów do komend itp.

WARUNKI

```
-- W1. Sprawdzenie czy jest plik
-- Brak
SELECT NOT EXISTS(
  SELECT doc_id FROM attachments WHERE doc_id =
  (SELECT doc_id FROM documents WHERE procid = {PROCID}))

-- Jest
SELECT EXISTS(
  SELECT doc_id FROM attachments WHERE doc_id =
  (SELECT doc_id FROM documents WHERE procid = {PROCID}))

-- W2. Sprawdzenie czy są zamieszczone opinie kierowników
-- pobierane jako parametry z bpm_properties.id_____ = 1
-- brak
SELECT count(*) = 0 FROM bpm_property_values
WHERE id_____ IN (12,13,14)
AND procid = {PROCID}
AND value_ != ''

-- są
SELECT count(*) > 0 FROM bpm_property_values
WHERE id_____ IN (12,13,14)
AND procid = {PROCID}
AND value_ != ''

W3.
-- czy dokument JEST dołączony do sprawy i powiązany z innym dokumentem (zakładka Powiązania)
SELECT dlp.prc_id IS NOT NULL AND dld.doc_id IS NOT NULL
FROM documents_view dv
LEFT JOIN doc_link_proc dlp USING(doc_id)
LEFT JOIN doc_link_doc dld ON dv.doc_id = dld.doc_id
WHERE dv.doc_id = {DOC_ID} LIMIT 1

W4.
-- Sprawdzenie czy dokument jest powiązany ze sprawą i dokumentem (zakładka powiązania)
SELECT dlp.prc_id IS NOT NULL AND dld.doc_id IS NOT NULL
FROM documents_view dv
LEFT JOIN doc_link_proc dlp USING(doc_id)
LEFT JOIN doc_link_doc dld ON dv.doc_id = dld.rel_to
WHERE dv.doc_id = {DOC_ID}

W5.
-- Sprawdzenie czy dokument ma wypełnione pozycje (np. zapotrzebowanie, faktura)
SELECT EXISTS (SELECT fk.adddat FROM fk_elements fk INNER JOIN documents d USING (doc_id)
WHERE d.doc_id = 50856 AND fk.is_del IS FALSE)

W6.
-- Sprawdzenie czy w sprawie zamknięte są wszystkie zapytania ofertowe (kontrolowane procedurą o prtpid = 5
SELECT sum(res) = 0 FROM
(SELECT
(CASE WHEN s.ptsttp = 'END' THEN 0 ELSE 1 END) AS res, d.dscrpt, s.ptstnm, d.prtpid
FROM documents d
INNER JOIN stages s ON d.is_del IS NOT TRUE AND s.procid = d.procid AND ((s.is_act IS TRUE AND s.is_fix IS FALSE) OR (s.pt
WHERE d.prtpid = 5 AND d.prc_id = {PRC_ID})) AS x
```

W7.

-- Sprawdzenie czy osoba zalogowana jest z określonego działu (wg orunid np. 57)

```
select substring(get_org_path(orunid), '.57.') IS NOT NULL FROM orgtree_view WHERE usr_id = {LOGGED_USR_ID}
```

W8.

-- Sprawdzenie wartości faktury z rozpisanymi kosztami

```
SELECT sum(amount) = (SELECT COALESCE(sum(netto_),0) FROM vatnote WHERE doc_id = {DOC_ID}) FROM vatnote_costs WHERE doc_id = {DOC_ID}
```

PRZYPISANIA

-- P1. Przypisanie jako osoby tworzącej dokument

```
SELECT ARRAY[o.orunid] FROM orgtree_view o INNER JOIN documents d ON d.adduid = o.usr_id
WHERE d.procid = {PROCID}
```

--

-- P2. Przypisanie akceptanta (pobierany z właściwości)

```
SELECT ARRAY[orunid] FROM organization_units WHERE orunid = {procedures.AKCEPTANT CZŁONEK ZARZĄDU}
```

--

-- P3. Przypisanie osób które zaakceptowały określony etap (np. 44)

```
SELECT ARRAY[orunid] FROM stages WHERE ptstid = 44 AND procid = {PROCID}
```

--

-- P4. Przypisanie osoby odpowiedzialnej ze sprawy

```
SELECT ARRAY(SELECT o.orunid FROM processes p INNER JOIN orgtree_view o ON p.rsuid = o.usr_id WHERE prc_id = {PRC_ID})
```

--

-- P5. Przypisanie osób z parametru typu usr_ids[]

```
SELECT ARRAY(SELECT o.orunid FROM orgtree_view o WHERE o.usr_id IN ({procedures.OSOBY}))
```

--

-- P6. Przypisanie osoby odpowiedzialnej za MPK wpisane w fakturze (do etapu MULTI)

```
SELECT ARRAY (
  SELECT CASE
    WHEN x.ndetpe = 'POST' THEN x.orunid
    WHEN x.ndetpe = 'ORGCELL' THEN (SELECT o2.orunid FROM orgtree_view o2 WHERE o2.prn_id = x.orunid LIMIT 1) END
  FROM
    (SELECT DISTINCT mpk.orunid, o.ndetpe FROM vatnote_costs
     LEFT JOIN places_of_vcosts AS mpk USING (povcid)
     LEFT JOIN orgtree_view AS o ON mpk.orunid = o.orunid
     WHERE doc_id = {DOC_ID})
  AS x)
```

--

-- P7. Przypisanie osoby zalogowanej

```
SELECT ARRAY(SELECT o.orunid FROM orgtree_view o WHERE o.usr_id = {LOGGED_USR_ID})
```

--

-- P8. Przypisanie orunid w zależności od accdid (jednostki org. w zależności od typu dokumentu księgowego)

```
SELECT
CASE WHEN v.accdid = 1 THEN [62]
      WHEN v.accdid = 2 THEN 56
      WHEN v.accdid = 3 THEN 61
      WHEN v.accdid = 4 THEN 60
      WHEN v.accdid = 5 THEN 63
```

```
END
```

```
FROM vatnote v WHERE v.doc_id = {DOC_ID}
```

```
-- P9. Przypisanie osoby, która załatwiła poprzedni etap
```

```
SELECT ARRAY[s.orunid]
```

```
FROM stages s
```

```
LEFT JOIN workflow_log wl USING(sop_id)
```

```
WHERE s.is_fix IS TRUE AND s.procid = {procedures.procid} AND wl.chloid = (SELECT max(chloid) FROM workflow_log WHERE proc
```

KWERENDY DO PARAMETRU SELECT[]

```
-- Członkowie zarządu
```

```
SELECT orunid as value, fullnm || ' - ' || ndenam as caption FROM orgtree_view WHERE orunid IN (3,14,15,16)
```

```
== DYNAMICZNE WARTOŚCI PARAMETRÓW KOMEND ==
```

```
-- Utwórz komentarz
```

```
dscrpt="SQL::SELECT CASE WHEN (SELECT EXISTS(
```

```
    SELECT value_
```

```
    FROM bpm_property_values WHERE id_____ = 20 AND sop_id = {SOP_ID})) THEN
```

```
(SELECT value_
```

```
    FROM bpm_property_values WHERE id_____ = 20 AND sop_id = {SOP_ID})
```

```
ELSE 'Bez uwag' END AS result
```

```
FROM stages WHERE sop_id = {SOP_ID}"
```

```
-- Utwórz przypomnienie w sprawie windykacyjnej
```

```
-- dla handlowca
```

```
usr_id="SQL::SELECT seller FROM vindication.vind_proc_view WHERE prc_id = {PRC_ID}",
```

```
start_="SQL::SELECT fxterm - interval '3 days' FROM vindication.vind_proc_view WHERE prc_id = {PRC_ID}",dscrpt="Uwaga! Za
```

```
--
```

```
-- Wszyscy pracownicy
```

```
SELECT orunid as value, fullnm || ' - ' || ndenam as caption FROM orgtree_view WHERE ndetpe = 'POST' AND is_del IS FALSE A
```

Zapytania do parametrów Akcji

Tworzenie dokumentu:

```
Tworzenie wydania z dokumentu przyjęcia
```

```
dctpid="17",dscrpt="Wydanie zewnętrzne",do="SQL::SELECT orunid FROM orgtree_view WHERE usr_id = {LOGGED_USER}"
```

```
# tworzenie zamówienia z zapotrzebowania
```

```
dctpid="41", dscrpt="Zamówienie do:{spller}",map="adddat=crtat,dlvdat=orddat,acrid=acrid,spadid=spadid,pchaid=pchaid",s
```

```
#przepisz pozycje z zapotrzebowania do zamówienia
```

```
from__="SQL::SELECT doc_id FROM documents WHERE doc_id={DOC_ID}", to__="SQL::SELECT doc_id FROM documents WHERE rel_to={
```

```
# przepisz pozycje z zapotrzebowania do istniejącego zamówienia
```

```
from__="SQL::SELECT doc_id FROM documents WHERE doc_id={DOC_ID}", to__="{procedures.ZAMÓWIENIE}"
```

```
# czy pozycje uzupełnione
```

```
query="SELECT (SUM(CASE WHEN depnam = depsym THEN 1 ELSE 0 END))=0 FROM fk_elements WHERE doc_id={doc_id} AND is_del=FALSE
```

```
# Utwórz sprawę (dsexid teczki, opis z opisu dokumentu, kontrahent z nadawcy, procedura o id 2
```

```
dsexid="107",dscrpt="{documents.dscrpt}",contid="SQL::SELECT contid FROM doc_link_cont WHERE doc_id = {DOC_ID} AND role__
```

```
# Zarejestruj dokument w zależności od tego czy dokument przyszedł z emaila
```

```
reg_id="
```

```

SELECT
CASE WHEN (SELECT EXISTS (SELECT e.doc_id FROM emails e WHERE e.doc_id = d.doc_id)) THEN 6
WHEN (SELECT NOT EXISTS (SELECT e.doc_id FROM emails e WHERE e.doc_id = d.doc_id)) THEN 4
END
FROM documents d WHERE d.doc_id = {DOC_ID}"

# Sprawdź czy workflow utworzonego z procedury dokumentu (o typie dctp_id 10) został zakończony
query="SELECT EXISTS(
SELECT doc_id FROM documents d INNER JOIN stages s ON s.procid = d.procid AND s.ptsttp = 'END'
AND d.doc_id = (SELECT dld.doc_id FROM documents d2 INNER JOIN doc_link_doc dld ON d2.doc_id = dld.doc_id WHERE d2.doc_id

# Sprawdź czy wypełnione są na opisie kosztowym konta grupy 4XX
SELECT EXISTS(SELECT substring(type__,1,1) = '4' AS res
FROM vatnote_costs INNER JOIN types_of_vcosts USING (tovcid)
WHERE doc_id = {DOC_ID} GROUP BY res HAVING substring(type__,1,1) = '4' )

# Wyślij email (do parametry Do - ustawienie emaili osób przypisanych do etapu)
SELECT text_sum(o2.e_mail) FROM
stages s3 INNER JOIN orgtree_view o2 ON o2.orunid = ANY(s3.orgarr)
WHERE s3.sop_id = {SOP_ID}

```