Wikiprint Book

Title: Przykłady SQL do raportów

Subject: Archiwum - eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - DeployerGuide/Custumization/AdvancedReporting/SQLExamples

Version: 31

Date: 09/09/26 02:53:00

Table of Contents

Przykłady SQL do raportów 3
Użyteczne konstrukcje i funkcje językowe SQL 3
Przykładowe użyteczne zapytania do bazy edokumenty 4

Przykłady SQL do raportów

Użyteczne konstrukcje i funkcje językowe SQL

```
--Formatuje datę
SELECT to_char(d.adddat, 'YYYY-MM-DD') FROM documents d;

-- Pobiera rok
select extract(YEAR from CURRENT_DATE);

-- Oblicza czas który upłynął od daty do daty (np. na realizacji zadania)
((extract(EPOCH FROM d.rlend_) - extract(EPOCH FROM d.rlstrt))/3600)::numeric(12,2) AS duration

-- Zamienia liczbę sekund na ciąg: H:M:S - np. 00:30:03
SELECT (1803 || 's')::interval

-- Zwraca kalendarz dni, nr tygodnia, miesiąć dla bieżącego roku
-- przydatne do JOIN-owania z zapytaniami
SELECT ndate, nday, to_char(ndate, 'D')::int - 1 AS dayweek, to_char(ndate, 'WW') AS week, to_char(ndate, 'MM') AS month F
(
SELECT sr.sr AS nday, (date_trunc('year', now()) + (sr.sr || 'day')::interval)::date AS ndate
FROM generate_series(0,365) AS sr
) AS calendar


-- Formatuje kwotę z pól tekstowych np. z formularzy customowych
select cast(regexp_replace(regexp_replace('301 110,43',' ',''),',', '.') as numeric(12,2)) + 12.50;

-- Formatuje kwotę zgodnie z polską normą
SELECT to_char(sum(netto_),'FM999 999 999') || ' PLN'
FROM vatnote v
INNER JOIN documents d USING(doc_id)
WHERE v.toctid = 3225 AND d.is_del IS FALSE
AND d.gostof IS NULL AND d.copyof IS NULL AND d.procid IS NOT NULL

-- Warunkowo koloruje pole
CASE WHEN p.pr_sta = 1 THEN '<div style="color: red">Rozpoczęte</div>' WHEN p.pr_sta = 2 THEN 'W trakcie realizacji' WHEN

-- Wybiera te rekordy których data jest w tym tugodniu
to_char(CURRENT_DATE, 'WW') = to_char (adddat, 'WW')

-- wybiera rekordy nie starsze niż miesiąc
AND dcr.adddat > CURRENT_DATE - interval '1 month'

-- ogranicza filtr jeśli {value} zwraca pusty string
CASE WHEN NULLIF('{value}', '') IS NULL THEN adduid IS NULL ELSE adduid IN ({value}) END


-- ogarnia puste wartości w parametrach (NULLIF)
SELECT doc.doc_id AS keyval, 'DOCUMENT' AS clsnam, doc.dscrpt,
to_char( doc.adddat, 'YYYY-MM-DD') AS datain
FROM documents doc
LEFT JOIN processes_view p USING (prc_id)
WHERE gostof is null AND doc.copyof IS NULL
AND doc.dscrpt !~ 'Powiadomienie: '
AND prc_id::text = (COALESCE(NULLIF('{PROCESSES_ID}','0')::text, '0'))::text
ORDER BY doc.adddat


-- Sprawdzenie wyników podzapytań w jednym zapytaniu (dla zachowania czytelności porównań)
```

```
SELECT CASE WHEN ile > tyle THEN 'OK' ELSE 'NIE' END FROM
(
SELECT (SELECT count(*) FROM files )  AS ile, 12 AS tyle
) AS x
```

Przykładowe użyteczne zapytania do bazy edokumenty

```
-- POBIERANIE DANYCH Z CECH
--
-- Przykład linkowania cech z dokumentu kastomowego o dctpid = 22
-- f18, f19, f20 to wartości pól tekstowe a f21 to lista JEDNOKROTNEGO wyboru
--
SELECT
doc.doc_id AS keyval,
'DOCUMENT'::text AS clsnam,
r.evdnum AS nr,
u.firnam || ' ' || u.lasnam AS imienazwisko,
CAST(doc.adddat AS date) AS datawniosku,
CAST(f18.data__ AS date) AS poczurlop,
CAST(f20.data__ AS date) AS konurlop,
CAST(COALESCE(f19.data__,'0') AS int) AS dniurlopu,
f21.ftopnm AS powod,
COALESCE(tops.dscrpt, '-') AS status
FROM documents doc
LEFT JOIN users u ON u.usr_id = doc.adduid
LEFT JOIN regofpapers r ON r.doc_id=doc.doc_id
LEFT JOIN features_text_view f18 ON doc.doc_id = f18.tbl_id AND f1.featid = 18
LEFT JOIN features_text_view f19 ON doc.doc_id = f19.tbl_id AND f19.featid = 19
LEFT JOIN features_text_view f20 ON doc.doc_id = f20.tbl_id AND f20.featid = 20
LEFT JOIN features_opt_view f21 ON doc.doc_id = f21.tbl_id AND f21.featid = 21
LEFT JOIN types_of_processes_states tops ON doc.tpstid = tops.tpstid
WHERE dctpid = 22 AND gostof IS NULL AND doc.is_del IS FALSE

--
-- Przykład sprawy z wartościami cech (tekstowe i opcje)
--
SELECT p.dscrpt, p.symbol, 'PROCESS'::text AS clsnam, prc_id AS keyval,
f1.data__ AS opinia, f2.ftopnm AS reklamacja, f3.data__ AS dzialanie_korygujace,
f4.data__ AS dzialanie_Klient, f5.data__ AS przyczyna
FROM processes_view p
LEFT JOIN features_text_view f1 ON p.prc_id = f1.tbl_id AND f1.featid = 14
LEFT JOIN features_opt_view f2 ON p.prc_id = f2.tbl_id AND f2.featid = 15
LEFT JOIN features_text_view f3 ON p.prc_id = f3.tbl_id AND f3.featid = 16
LEFT JOIN features_text_view f4 ON p.prc_id = f4.tbl_id AND f4.featid = 17
LEFT JOIN features_text_view f5 ON p.prc_id = f5.tbl_id AND f5.featid = 18
WHERE p.prtpid = 1

---
--- Przykład czasu wykonania etapów procedury o ID 28 i 29
---
SELECT sd.ptstnm, avg(s.fixdat - s.actdat) AS avarage, (extract(epoch FROM avg(s.fixdat - s.actdat))/3600)::int AS avhours
FROM stages s
INNER JOIN stages_def sd USING(ptstid)
INNER JOIN procedures pr ON (pr.procid = s.procid OR s.procid = ANY (pr.pararr))
INNER JOIN documents d ON (d.procid = pr.procid OR d.procid = pr.rootpr)
WHERE d.is_del IS FALSE AND d.gostof IS NULL
AND s.is_fix = TRUE
AND s.actdat IS NOT NULL
AND sd.prtpid IN (28, 29)
AND sd.is_del IS FALSE
```

```
AND sd.ptsttp IN ('ACTIVITY')
GROUP BY sd.ptstnm, s.ptstid, sd.ptsttp
ORDER BY avarage DESC



--
-- Przykład pobrania cechy kontrahenta - listy WIELOKROTNEGO wyboru
-- konieczne jest użycie funkcji agregującej text_sum aby nie otrzymywać podwójnych rekordów
--
SELECT c.name_1, text_sum(f2.ftopnm)
FROM contacts c
LEFT JOIN features_opt_view f2 ON c.contid = f2.tbl_id AND f2.featid = 20
WHERE contid = {CONTID}
GROUP BY c.name_1

--
-- Przykład linkowania cech zdarzeń
--
SELECT to_char(d.adddat, 'YY-MM-DD') AS day,
rozmowy
(SELECT count(*) /10
FROM events_view e WHERE e.trmtyp = 'PHONECALL' AND emp_id IN (84,62) AND e.start_ >= d.adddat::date - 7
AND e.start_ <= d.adddat) AS rozmowy_handlowe,
spotkania
(SELECT count(*)
FROM events_view e WHERE e.trmtyp = 'MEETING' AND emp_id IN (84,62) AND e.start_ >= d.adddat::date - 7
AND e.start_ <= d.adddat) AS spotkania,
wartość pola featid 98
f3.data__::int AS odwedok,
f4.ftopnm AS cecha4
FROM documents d
INNER JOIN features_text_view f3 ON d.doc_id = f3.tbl_id AND f3.featid = 98
cecha z listy wyboru
INNER JOIN features_opt_view f4 ON d.doc_id = f4.tbl_id AND f4.ftopid = 119
WHERE d.is_del IS NOT true AND d.gostof IS NULL



--
-- Przykład listy spraw z cechami - tekstowymi i listami
--
SELECT
-- Wybieramy klucz główny i nazwę formularza do podłączenia
--(ustawić w polach typ i wartość raportu - aby otwierać obiekty)
p.prc_id as keyval, 'PROCESS'::TEXT as clsnam,
-- opis sprawy i inne pola z tabeli lub jeszcze lepiej widoku processes_view
dscrpt,
-- liczba porządkowa
ROW_NUMBER() over (order by p.prc_id) AS lp,
-- klient nazwa krótka jesli jest, jeśli nie to pełna
COALESCE(c.name_2, c.name_1) AS klient,
-- w zależności  od terminu - kolorowanie
CASE WHEN fxterm > CURRENT_DATE THEN
 '<div style="background-color: ' || fxterm  || '; width: 100%; height: 100%">' || fxterm || '</div>'
ELSE '' || fxterm END AS fxterm,
f40.data__ AS moja_cecha_tekstowa,
f31.ftopnm AS moja_cecha_lista
FROM processes_view p
LEFT JOIN contacts c USING (contid)
LEFT JOIN features_text_view f40 ON f40.tbl_id = p.prc_id AND f40.featid = 40
LEFT JOIN features_opt_view f31 ON f31.tbl_id = p.prc_id AND f31.featid = 31
```

```
-- tylko z teczki o podanym dsexid (mozna sprawdzić na wyciągu z wykazu akt)
AND p.dsexid IN (646)
WHERE p.is_del IS FALSE




-- Wybiera symbol teczki z numeru sprawy oo formacie ''2/03/08/UP/AW''

SELECT substring(symbol from '[0-9]*/[0-9]*/[0-9]*/([A-Z]*)') FROm processes;


-- Wybiera krótką nazwę klienta jeśli jest, a jeśli jest pusta to długą

SELECT COALESCE(c.name_2, c.name_1) FROM contacts c;

-- Wybiera dane do trendu - do wykresu

SELECT extract(month from rlstrt) as m,
sum((time__::numeric(12,2)/3600)::numeric(12,2)) AS sum
FROM rcp_cards_view
WHERE rlstrt IS NOT NULL AND is_fak = TRUE AND is_del = FALSE AND tpstid = 9
AND rlstrt + interval '1 year' >= CURRENT_DATE
GROUP BY extract(year from rlstrt),
extract(month from rlstrt)
ORDER BY extract(year from rlstrt), extract(month from rlstrt)

-- Formatuje status z obrazkiem czy zakończony
-- Pokazuje wiele osób na równoegłych etapach w obiegu faktury
SELECT 'DOCUMENT'::text AS clsnam, d.doc_id AS keyval, d.doc_id, v.amount, v.vat___, v.number, reg.evdnum,
regexp_replace(d.dscrpt, 'Od:[ 0-9a-zA-ZąęłóśćńźżĄĘŚÓĆŃŁŹŻ\";\./.)(,-]*$', '')  AS dscrpt, d.trgtxt, v.payed_,  v.seldat::
CASE WHEN tps.status IN ('FINAL', 'CANCEL') THEN '<img src=../../framework/img/MessageBox/checked.png width=18 title=Statu
v.paytrm::date AS paytrm,
COALESCE(name_1, name_2) AS contrah,
ftv.ftopnm AS rodzaj,
tps.dscrpt AS status,
tps.tpstid,
toa.code__,
(SELECT text_sum(osoby) FROM (
SELECT DISTINCT (firnam||' '||lasnam) AS osoby
FROM stages s
INNER JOIN procedures pr2 USING(procid)
LEFT JOIN orgtree_view o3 ON o3.orunid = any(s.orgarr)
WHERE
  ((s.is_act IS TRUE AND s.is_fix IS FALSE)
   OR (s.ptsttp = 'END' AND s.is_fix IS TRUE))
   AND (s.ptsttp != 'SUBPROCESS') AND (pr2.rootpr = d.procid)
) x) AS osoby,
d.procid,
pr.prtpnm
FROM documents d
INNER JOIN vatnote v USING(doc_id)
INNER JOIN contacts c ON (c.contid = v.frctid)
LEFT JOIN regofvatnotes reg USING (doc_id)
LEFT JOIN features_opt_view ftv ON d.doc_id = ftv.tbl_id AND ftv.featid = 2
LEFT JOIN types_of_accountants_doc toa USING(accdid)
LEFT JOIN types_of_processes_states AS tps ON d.tpstid = tps.tpstid
LEFT JOIN procedures pr USING(procid)

WHERE d.gostof IS NULL AND d.copyof IS NULL AND d.state_ = 2 AND d.is_del IS FALSE
AND d.adddat BETWEEN '{DATE_FROM}'::date AND '{DATE_TO}'::date
```

```
AND {FILTER_STRING}
ORDER BY d.doc_id

--
-- Tworzy listę spraw z liczbą porządkową
-- Liczba porządkowa pochodzi z sekwencji
--
DROP sequence if exists lp;
CREATE temp sequence lp start 1;

SELECT nextval('lp') AS lp, dscrpt, symbol, dsexid, optime, cltime, fullnm, name_1
FROM processes_view
WHERE is_del IS FALSE
LIMIT 100

--Pozyskiwanie prc_id spraw nadrzędnych w formie kolumny

SELECT unnest((SELECT pararr FROM processes WHERE prc_id = {PRC_ID}))

--Pozyskanie prc_id sprawy głównej nadrzędnej

SELECT prc_id
FROM processes
WHERE prc_id IN ((SELECT unnest(
(SELECT pararr
FROM processes WHERE prc_id = {PRC_ID})))
UNION
(SELECT {PRC_ID}))
AND rel_to IS NULL

--Pozyskiwanie prc_id wszystkich spraw, które znajdują się pod sprawą główną (nadrzędną) dla sprawy z {PRC_ID}
--Zapytanie zwróci nam prc_id wszystkich spraw zgodnie ze strukturą niżej
--sprawa główna
----sprawa podrzędna nr1
------sprawa podrzędna do sprawy nr1 - {PRC_ID} z tego poziomu startujemy
----sprawa podrzędna nr2

SELECT prc_id
FROM processes WHERE (SELECT prc_id
FROM processes
WHERE prc_id IN ((SELECT unnest(
(SELECT pararr
FROM processes WHERE prc_id = {PRC_ID})))
)
AND rel_to IS NULL) = ANY(pararr)
UNION
(SELECT prc_id
FROM processes
WHERE prc_id IN ((SELECT unnest(
(SELECT pararr
FROM processes WHERE prc_id = {PRC_ID})))
)
AND rel_to IS NULL)

--
-- Lista spraw wraz z kolorowaniem statusy
--
drop sequence if exists lp;
create temp sequence lp start 1;
SELECT nextval('lp') AS lp, * FROM
(SELECT p.prc_id as keyval, 'PROCESS'::TEXT as clsnam, COALESCE(p.dscrpt, '-') AS dscrpt, p.c_name, p.optime::date, c.name
```

```
p.fxterm, p.foreda::date, p.forepa, p.credib, p.rspini,
p.lm_dat::date AS ost_mod,
(CASE WHEN p.currpa != 'PLN' THEN
(SELECT cr_avg FROM currency_exchange_table cet WHERE cet.cursmb = p.currpa ORDER BY crdate LIMIT 1)
ELSE 1 END) * p.forepa AS wartosc_wal,
p.currpa AS waluta,
'<div style="background-color: ' || tops.colour || '; width: 100%; height: 100%; padding: 2px">' || tops.dscrpt || '</div>
FROM processes_view p
LEFT JOIN contacts c USING(contid)
LEFT JOIN types_of_processes_states tops ON (tops.state_::int = p.pr_sta AND tops.clsnam = 'PROCESS')
WHERE p.is_del IS FALSE AND p.is_fix IS false
AND rspuid = {LOGGED_USR_ID}
AND {FILTER_STRING}
AND forepa > 0
ORDER BY tops.prior_, p.forepa DESC) AS x


-- Select listy dokumentów do których uprawniona jest grupa użytkownika
SELECT * FROM documents_view WHERE is_del IS NOT TRUE AND (SELECT prtpid = 11)
AND doc_id IN (
 SELECT doc_id FROM doc_link_users dlu
 WHERE dlu.grp_id IN (SELECT grp_id FROM users_link_group glu WHERE glu.usr_id = 2)
 AND dlu.doc_id = doc_id
)
```