Wikiprint Book

Title: Tworzenie raportów w SQL

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

Version: 51

Date: 08/06/26 11:20:46

Table of Contents

Tworzenie raportów w SQL 3
- *Instrukcja tworzenia nowego raportu* 3
- *W kwerendach można stosować parametry* 3
- *Konfiguracja automatycznych raportów* 3
- *Umożliwienie otwierania dialogów (formularzy) z wyników raportu* 4
  - *Otwieranie innych obiektów za pomocą linku "Formatu link do"* 6
- *Wykresy* 6
- *Grupowanie, sumowanie i sortowanie* 6
- *Menu raportów* 6
- *Drill down* 8
- *Raporty z systemu ERP* 8
- *Biblioteka raportów* 8
- *Filtry dla raportów* 8
- *Tips & Tricks* 9
- *Użyteczne komendy SQL* 9

Tworzenie raportów w SQL

Do tworzenia zaawansowanych raportów SQL przydatna będzie oprócz PGAdmina również dokumentacja bazy danych. Znajduje się ona Tutaj. Jest możliwa również do pobrania jako plik DokumentacjaBazyDanycheDokumenty.zip.

Instrukcja tworzenia nowego raportu

1. Przechodzimy do eDokumentów, moduł Raporty > Nowy raport (min. Nazwa, Grupa) - zapisujemy, otwierają się dodatkowe zakładki
1. W zakładce Definicja wpisujemy Kwerendę SQL "SELECT * FROM events" (UWAGA! Zapytanie musi zwracać przynajmniej 1 rekord)
1. Zapisujemy, Otwieramy do edycji ponownie. Na zakładce Wybór kolumn przenosimy klikając w przyciski te pola które chcemy wyświetlić.
1. Zapisujemy raport.
1. Otwieramy raport aby obejrzeć wyniki

W kwerendach można stosować parametry

```
 {DATE_FROM} (string) - np. adddat::date >= '{DATE_FROM}'
{DATE_TO} (string) - np. adddat:: <= '{DATE_TO}'
{USR_ID} (string) - przecinkami rozdzielona lista użytkowników do którym ma dostęp zalogowany user
{LOGGED_USR_ID} (int) - id zalogowanego pracownika,
{ENT_ID} (int) - id jednostki na której jest wykonywany raport
{SYMBOL} (string) - symbol kontrahenta z systemu zewnętrznego
{EXTEID:OPTIMA}(int) - symbol kontrahenta z systemu Optima
{ACORID} (int) - id jednostki rozliczeniowej
{TOVCID} (int) - id rodzaju kosztów
{CONTID} (int) - id kontrahenta (przy raportach wykonywanych w kartoteki)
{CONTIDS} (int[]) - id kontrahentów zaznaczonych na liście w module Klienci
{DOC_ID} (int) - id dokumentu (przy raportach wykonywanych w kartoteki)
{DOC_IDS} (int[]) - id dokumentów zaznaczonych na liście dokumentów
{DOCIDS} to samo co {DOC_IDS} zaleca się używania {DOC_IDS}
{PRC_ID} (int) - id sprawy (przy raportach wykonywanych w kartoteki)
{PRC_IDS} (int[]) - id spraw zaznaczonych na liście spraw
{PRCIDS} to samo co {PRC_IDS}
{EVNTID} (int) - id zdarzenia (przy raportach wykonywanych w kartoteki)
{EVNTIDS} (int[]) - id zdarzeń zaznaczonych na liście
{CAMPID} (int) - id kampanii (moduł Kampanie -> zaznaczona kampania na drzewku kampanii)
{DEVCID} (int) - id urządzenia (przy raportach wykonywanych w kartoteki)
{DEVCIDS} (int[]) - id urządzeń zaznaczonych na liście
{DEPOID} (int) - id produktu
{DEPOIDS} (int[]) - id produktów zaznaczonych z listy produktów
{FILTER_STRING} (string) - wartość zwrócona przez filtr
```

będą one mapowane na formularzu "Parametry raportu" i z odpowiednich pól pobierane będą wartości.

Całe zapytanie raportu jest dodatkowo objęte przepuszczane przez silnik parsujący (jak np. w workflow). Co daje dodatkowe możliwości (np. dynamiczne parametry w raportah wykonywanych na zew. źródłach danych). Przykładowe zapytanie z raportu wykonywanego z kontekstu dokumentu:

```
SELECT * FROM mssql_external_documents WHERE symbol = '{SQL::SELECT ex_sym FROM documents WHERE doc_id = {DOC_ID}}'
```

Konfiguracja automatycznych raportów

Jeżeli chcielibyśmy otrzymywać regularnie jakiś raport "na biurko" mamy taką możliwość poprzez funkcjonalność planowania raportów. W efekcie np. raz w tygodniu w piątek otrzymamy raport wysłanych ofert w formacie PDF na naszą skrzynkę dokumentów.

Dowolny raport można dodać do zaplanowanych na zakładce "Planowanie" formularza edycji raportu.

Ponieważ można dodać tylko jedno zadanie dla jednego raportu dlatego nie jest możliwa edycja zadania. Zawsze nowo-wstawiony rekord nadpisze poprzedni.

Aby raporty wykonywane były automatycznie należy upewnić się że skrypt CronRunner.php jest dodany do zaplanowanych zadań systemu operacyjnego.

W systemie Linux można dodać go poprzez skopiowanie pliku cronrunner do katalogu /etc/cron.d/ lub poprzez edycję pliku /etc/cron.d/crontab.

```
*/1 * * * * www-data cd /home/edokumenty/public_html/apps/edokumenty && php -f CronRunner.php >> /var/log/cronrunner.log

schtasks /create /sc co_minutę /mo 5 /tn "CronRunner" /tr "php -f C:\Program Files\BetaSoft\eDokumenty\CronRunner.php"

(Na serwerach modyfikator "sc co_minutę" należy zastąpić słowem angielskim "/sc minute")
```

Ponieważ skrypt tworzący plik PDF generuje dużo ostrzeżeń, jeżeli nie chcemy debugować skryptu nie jest zalecane przekierowywanie wyników działania do maila (opcja MAILTO powinna być wyłączona)

Umożliwienie otwierania dialogów (formularzy) z wyników raportu

Tworzymy zapytanie które w kolumnach o nazwach *clsnam* i *keyval* będą zawierać odpowiednio NAZWĘ_FORMULARZA i wartość klucza podstawowego rekordu np.

```
SELECT
'PROCESS' AS clsnam,
prc_id AS keyval,
dscrpt FROM processes;
```

Następnie na zakładce *Definicja* raportu wpisujemy aliasy tych pól.

- Typ z pola: clsnam
- ID z pola: keyval

Dostępne formularze i ich klucze podstawowe:

Klasa	Klucz	Kartoteka/formularz
PROCESS	prc_id	Sprawa
DOCUMENT	doc_id	Dokument
CONTACT	contid	Klient
RCP	rcp_id	Karta RCP
MEETING	evntid	Spotkanie
TODO	evntid	Zadanie
PHONECALL	evntid	Rozmowa telefoniczna

EVENTNOTE	evntid	Adnotacja do zdarzenia
ALARM	evntid	Przypomnienie
CORPEVENT	evntid	Wydarzenie korporacyjne
FILE	fileid	Właściwości pliku (załącznika)
FKPROCESSELEMENT	fkelid	Pozycja sprawy
FKRCPELEMENT	fkelid	Pozycja RCP
FKDEMANDELEMENT	fkelid	Pozycja zapotrzebowania
FKPZELEMENT	fkelid	Pozycja przyjęcia zewnętrznego
FKWZELEMENT	fkelid	Pozycja wydania zewnętrznego
FKORDERELEMENT	fkelid	Pozycja zamówienia
FKOFFERELEMENT	fkelid	Pozycja oferty
FKVATNOTEELEMENT	fkelid	Pozycja faktury VAT
DEVICE	devcid	Urządzenie (moduł serwis)
CAMPAIGN	capmid	Kampania
PRODUCT	depoid	Produkt z cennika
PROJECT	projid	Projekt

CONTACTPERSON	copeid	Osoba kontaktowa
CREGISTER	id	Wpis rejestru

Ważne jest aby używać odpowiednich klas dla danych obiektów. Szczególnie ważne jest to w przypadku pozycji (produkty) na dokumentach FK oraz sprawie gdzie klasa jest odpowiedzialna na otwarcie konkretnego formularza a co za tym idzie logiki biznesowej.

Otwieranie innych obiektów za pomocą linku "Formatu link do"

W zakładce Wybór kolumn dla kolumny która ma stać się linkiem można użyć funkcji formatującej. Aby otworzyć dialog kolumna musi zwracać ciąg zawierający elementy "CLSNAM|KEYVAL|TEXT" np:

```
CONTACT|3241|Otwórz kontakt
PROCESS|5211|Otwórz sprawę
CREGISTER|132144|Otwórz
```

Wykresy

Dostępna jest możliwość drukowania wykresów, należy jednak odpowiednio sformułować zapytanie, tak aby wyniki możliwe były do wyświetlenia na wykresie: słupkowym, liniowym i kołowym (Pie).

Aby wydrukować wykres kołowy jego definicja musi być tak ułożona, aby pierwsza kolumna wskazywała na opis a druga na wartość !!

Grupowanie, sumowanie i sortowanie

Raport będzie sortowany wg kolejności pól podanych w definicji raportu po wyrażeniu kluczowym "order by". Przykład :

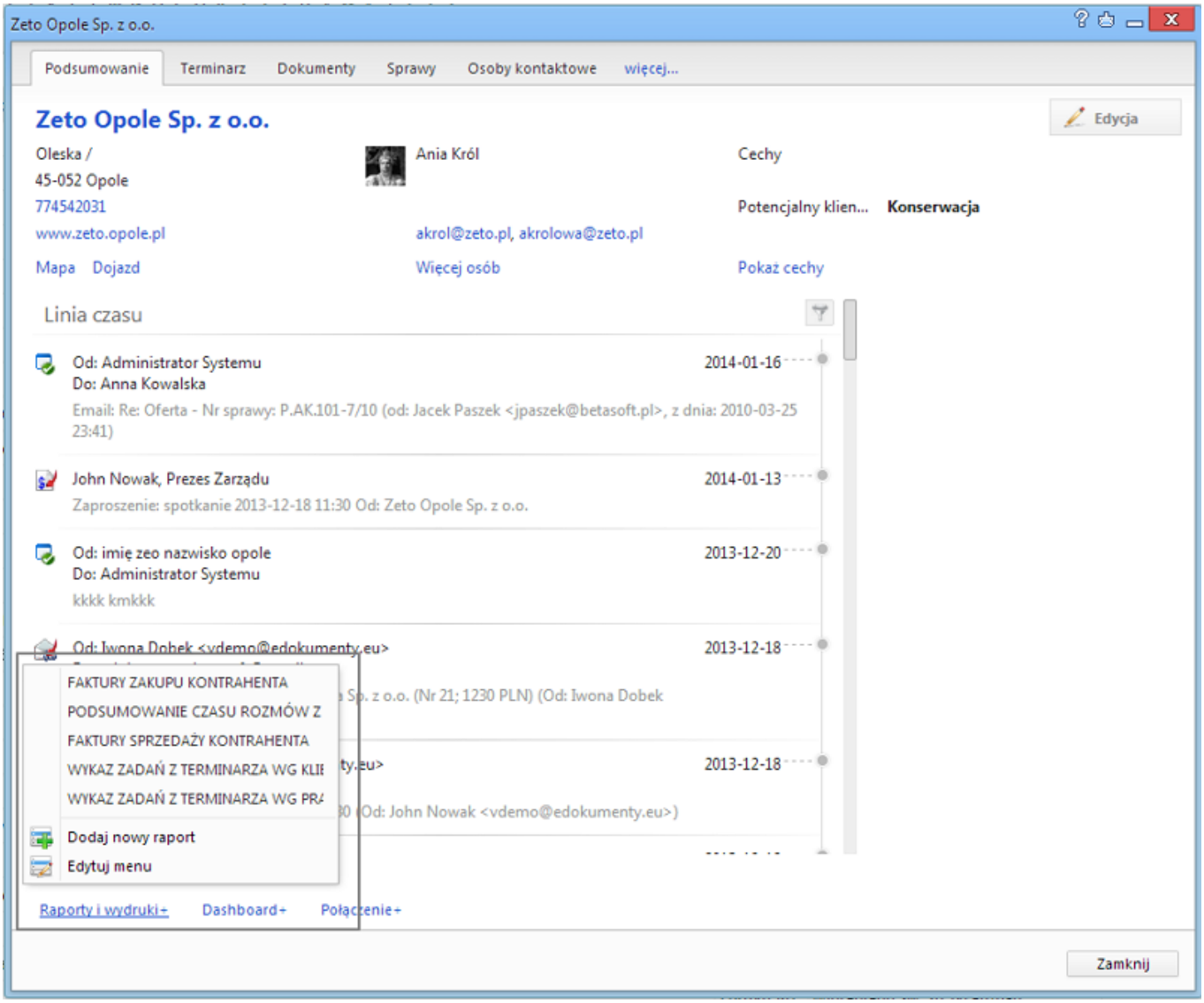
```
SELECT name_1, name_2, nip___ ,ph_num  FROM contacts ORDER BY name_1
```

Raport może zawierać grupowanie po 1 wybranym polu - należy je wpisać do sekcji "Grupowanie" na zakładce "Definicja" edytora raportów. Pole to należy uwzględnić w klauzuli "order by" w samej definicji raportu.

Możliwe jest sumowanie wartości poszczególnych kolumn - w tym celu należ pola te wymienić po przecinkach w sekcji "Sumowanie".

Menu raportów

Wybrane raporty można udostępnić w formularzu sprawy lub klienta. Będą dostępne na dole kartoteki pod przyciskiem *Raporty*.



Rys. 1

Aby dodać raport można użyć przycisków *Dodaj raport* (Rys. 1), można również to zrobić w Ustawieniach. W tym celu należy wybrać Ustawienia -> Menu raportów. W polu "Nazwa dialogu" wybieramy nazwę formatki do której chcemy dodać raport najczęściej będą to kartoteka klienta lub kartoteka sprawy. W liście wyboru ustawiamy interesujący nas raport. W definicji wywoływanego w ten sposób raportu używamy odpowiednio parametru {CONTID} albo {PRC_ID} - aby raport użył otwartej kartoteki jako parametru.

Takie raporty można również wywoływać na zewnętrznych źródłach danych. Warunkiem jest posiadanie identyfikatora kontrahenta z zewnętrznego systemu. Widać je na zakładce Kontakt kartoteki klienta. W kwerendzie SQL odwołujemy się do nich:

```
{EXTEID:OPTIMA}(int) - symbol kontrahenta z systemu Optima
{SYMBOL} (string) - symbol kontrahenta z systemu zewnętrznego
```

OPTIMA - to jest nazwa systemu, który jest zintegrowany - widać ją również w liście wyboru.

Dodatkowo od wersji 4.4 będzie można wykonywać zapytania do systemu zewnętrznego wykorzystując inne dane niż klucz systemu zewnętrznego w jednym raporcie:

```
SELECT * FROM order_ext o LEFT JOIN customer c ON c.id = o.customer_id
WHERE o.end = '0'
AND nip = '{SQL::SELECT c.nip___ FROM contacts c INNER JOIN vatnote v ON (v.frctid = c.contid) WHERE v.doc_id = {DOC_ID}}'
```

Sekwencja `'{SQL::SELECT ...'` jest zapytaniem do bazy danych systemu eDokumenty, natomiast zapytanie nadrzędne jest zapytaniem do bazy danych systemu zewnętrznego

Drill down

Bardzo użytecznym mechanizmem jest możliwość zagłębiania raportów w sobie, dzięki czemu można ujrzeć dokładniej skąd pochodzą sumowane w kolumnach dane.

Aby utworzyć raport drill-down należy w kolumnie którą chcemy rozwijać dodać odpowiedni parametr. Ponieważ lista parametrów może być rozwijana w przyszłości zastosowano tam składnię JSON. Przykładowo aby rozwinąć dane o miesięcznej sprzedaży do sprzedaży wg produktów (reprezentowanej przez raport o ID 196) należy wkleić do komórki PARAMETRY danej kolumny na zakładce "Wybór kolumn":

```
{"drilldown":{"rep_id":196}}
```

a aby wpodrzędnym raporcie filtrować odpowiednie wpisy, można odwoływać się do krotki raportu nadrzędnego w formie {NAZWA_KOLUMNY}. Działa to również dla wielu kolumn, więc warunek może być łączony np. : WHERE to_char(v.seldat, 'YYYY-MM') = '{MON}' AND mpk.place_ = '{MPK_PLACE}' gdzie *mon* i *mpk_place* są kolumnami w raporcie nadrzędnym.

Demo Sp. z o.o. Drukuj

7. SPRZEDAŻ MIESIĘCZNIE z rozwinięciem na podział wartości wg produktów

Opis:
Data utworzenia: 2012-09-24
Użyte parametry:
Akcje: Określ parametry Eksportuj do XLS Eksportuj do PDF Zapisz w systemie

Miesiąc	Wartość
⊞ 2012-03	47 340.80
⊞ 2012-04	17 500.00
⊞ 2012-05	250.00
⊟ 2012-07	196 056.67

Produkt	Wartość
Zepół Pomp	149 970.00
iRobot	36 000.00
Canon DR-2510C	6 635.16
Windows 7	2 500.00
Nagrywarka zew, DVD FREECOM FS DL Lightscribe 8x USB FireWire	951.51

⊞ 2012-08	58 996.31
⊞ 2012-09	18 900.00

Bardziej zaawansowane parametry i weryfikacja JSON: http://json.parser.online.fr/

Raporty z systemu ERP

Podobnie możemy w systemie zdefiniować raporty obrazujące dane zgromadzone w dowolnym systemie ERP np. koncentrację sprzedaży, trend sprzedaży, poziom kosztów, zamówienia, stany magazynowe itp.

Wystarczy że w definicji raportu odwołamy się do zewnętrznego źródła danych i wpiszemy odpowiednie kwerendy SQL.

Biblioteka raportów

Raporty z biblioteki można pobierać poprzez przeglądarkę lub klienta webdav z serwera support.

Każdy katalog zawiera jeden raport, każdy raport reprezentowany jest przez co najmniej 3 pliki.

- *Zestawienie spotkan.report* - plik raportu do importu
- *Spotkania.sql* - może zawierać również dodatkowe kwerendy ale wyraźnie oddzielone od właściwej oraz komentarze
- *Spotkania.png* - screenshot dla łatwiejszej orientacji
- opcjonalnie pliki HTML dla raportów z szablonami np. *szablonAudytu.html*

Filtry dla raportów

Filtry dla raportów

Tips & Tricks

Dla łatwiejszego tworzenia raportów można użyć narzędzia PgAdmin, wówczas dla sieci lokalnej konfiguracja pg_hba.conf powinna wyglądać dla sieci w której serwer ma adres przykładowo:

```
[root@edokumenty ~]# ip a | grep eth0
inet 10.8.16.33/24 brd 10.8.16.255 scope global eth0

# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
host    oblig       all         10.8.16.255/24        trust
```

W postgresql.conf należy ustawić nasłuchiwanie na wszystkich interfejsach sieciowych.

```
listen_addresses = '*'
```

Po skończeniu wdrożenia koniecznie przywrócić do pierwotnej postaci.

Użyteczne komendy SQL

Przykłady użytecznych kwerend Przykładowe raporty?