Podręcznik Administratora

# Integracja SIMPLE

## Integracja z SIMPLE

W bazie danych istnieją zdefiniowane widoki: simple\_pozycja\_dok – zawiera pozycje dokumentów ruchów magazynowych simple\_faktura, simple\_pozycja – faktury, pozycje faktur simple\_kontrahent – kontrahenci simple\_indeks\_jm – indeks TiU

Aby stworzyć możliwość korzystania z wymienionych widoków należy wykonać poniższe kroki

### Założenie użytkownika, nadanie praw do połączenia z bazą danych, sprawdzenie logowania

-na serwerze bazy danych, korzystając z konsoli logujemy się do bazy danych:

psql NAZWA\_BAZY -U postgres (lub root) -W

Poprawne dane dot. użytkownika oraz serwera znajdują się w pliku build.properties w katalogu głównym aplikacji.

-tworzymy użytkownika bazy danych:

create user czytelnik with password 'czytelnik'; -nadajemy prawa łączenia się do bazy danych:

grant connect on database "NAZWA\_BAZY" to czytelnik;

-testujemy połączenie

psql -U czytelnik -W -c "select \* from simple\_faktura limit 1;" NAZWA\_BAZY

### Zmiana ustawień postgresql - włączenie nasłuchiwania na określonym adresie

aby serwer bazy danych oczekiwał na połączenia na danej karcie sieciowej należy zmodyfikować plik: postgresql.conf znajdujący się w katalogu /etc/postgresql/[wersja postgres'a]/main/ Przed modyfikacją pliku należy wykonać jego kopię.

W części CONNECTIONS AND AUTHENTICATION należy odkomentować opcję listen\_addresses oraz zmienić jej wartość zgodnie z instrukcją w komentarzu.

Np jeśli adres serwera w sieci lokalnej to 192.168.56.1 i to właśnie z tej sieci będzie potrzeba dostawania się do widoków to należy dodać ten adres do wartości zmiennej listen\_addresses:

listen\_addresses = 'localhost,192.168.56.1' Jeśli zmienna listener\_addresses była już odkomentowana i miała ustawione wartości to należy sprawdzić, czy wśród wartości był adres, który chcemy ustawić i ew. dodać go.

Po zmodyfikowaniu postgresql.conf należy zrestartować posgresql'a - dzięki temu zobaczymy, czy np nie popełniliśmy błędów w pliku konfiguracyjnym - jeśli plik będzie zawierał błędy to postgresql nie uruchomi się.

### Zmiana ustawień postgresql - dodanie adresów z których może następować łączenie do bazy danych

aby była możliwość przyjęcia przez postgresql połączeń z danej sieci należy zmodyfikować plik: pg\_hba.conf znajdujący się w katalogu /etc/postgresql/[wersja postgresql]/main/ Przed modyfikacją pliku należy wykonać jego kopię. Na końcu pliku należy dodać linie:

host all all 192.168.56.0 255.255.255.0 password

192.168.56.0 255.255.255.0 należy zastąpić adresem sieci oraz maską sieci z której będą następowały połączenia.

Po zmodyfikowaniu pg\_hba.conf należy zrestartować posgresql - dzięki temu zobaczymy, czy np nie popełniliśmy błędów w pliku konfiguracyjnym - jeśli plik będzie zawierał błędy to postgresql nie uruchomi się.

### Test połączenia

na komputerze z którego użytkownik będzie ”dostawał” się do widoków należy zainstalować postgresql. Aby przetestować połączenie z konsoli należy wydać polecenie:

psql -U czytelnik -W -h ADRES\_IP\_SERWERA\_Z BAZA\_DANYCH -c "select \* from simple\_faktura limit 1;"

NAZWA\_BAZY po podaniu hasła powinniśmy zobaczyć wyświetlony jeden wiersz widoku simple\_faktura.

## Dostępne widoki bazodanowe

### Ogólne

**Widok simple\_indeks\_jm** zwraca informacje dotyczące indeksu towarów i usług wraz z jednostkami miar oraz grupą towarową.

Zapytanie:

|  |
| --- |
| SELECT jm.jm\_id AS indeks\_jm\_id, jm.jm\_kod AS indeks\_jm\_kod, indeks.indeks\_nazwa, jm.jm\_nazwa, indeks.sww, indeks.pkwiu, indeks.stawka\_vat, indeks.indeks\_kod, grupa.grupa\_kod, grupa.nazwa\_z\_hierarchia AS grupa\_pelna\_nazwa FROM jm JOIN indeks ON indeks.indeks\_id = jm.indeks\_id LEFT JOIN grupa ON grupa.grupa\_id = indeks.grupa\_id ORDER BY indeks.indeks\_nazwa, jm.jm\_nazwa; |

Opis kolumn:

indeks\_jm\_id - identyfikator jednostki miary, indeks\_jm\_kod - kod jednostki miary, indeks\_nazwa - nazwa indeksu, jm\_nazwa - nazwa jednostki miary, sww - symbol SWW, pkwiu - symbol PKWiU, stawka\_vat - stawka VAT, indeks\_kod - kod indeksu, grupa\_kod - kod grupy, grupa\_pelna\_nazwa - nazwa grupy.

Przykładowe wykorzystanie:

SELECT \* FROM simple\_indeks\_jm;

Wynik:



**Widok simple\_kontrahent** zwraca informacje dotyczące kontrahentów.

Zapytanie:

SELECT instytucja.instytucja\_id AS kontrahent\_id,

"substring"(instytucja.nazwa::text, 1, 60) AS nazwa, "substring"(instytucja.nazwa::text, 1, 80) AS nazwa\_rozw, btrim(instytucja.nr\_domu::text) || CASE WHEN instytucja.nr\_lokalu IS NOT NULL THEN '/'::text || btrim(instytucja.nr\_lokalu::text) ELSE *::text*  END AS nrdomu, instytucja.ulica, instytucja.miejscowosc AS miasto, instytucja.kod\_pocztowy AS kodpoczt, instytucja.instytucja\_id AS odbiorca\_idn,

replace(instytucja.nip::text, '-'::text, *::text) AS nip*  FROM instytucja;

Opis kolumn:

kontrahent\_id - identyfikator kontrahenta, nazwa - nazwa kontrahenta, nazwa\_rozw - pełna nazwa kontrahenta, nrdomu, ulica, miasto, kodpoczt - składają się na adres kontrahenta, odbiorca\_idn - identyfikator, nip - NIP.

Przykładowe wykorzystanie:

SELECT \* FROM simple\_kontrahent;

Wynik:



### Faktury

**Widok simple\_faktura**

zwraca informacje dotyczące faktur sprzedaży oraz paragonów i kwitów kasowych Zapytanie:

|  |
| --- |
| CREATE OR REPLACE VIEW simple\_faktura AS SELECT  CASE WHEN faktura.typ\_faktury::text = 'W'::text AND faktura.kupujaca\_instytucja\_id IS NULL THEN 'KOMERCJA'::text WHEN faktura.typ\_faktury::text = 'F'::text AND faktura.kupujaca\_instytucja\_id IS NOT NULL AND instytucja.typ\_instytucji::text = 'NFZ'::text THEN 'NFZ'::text WHEN faktura.typ\_faktury::text = 'F'::text AND faktura.kupujaca\_instytucja\_id IS NOT NULL AND instytucja.typ\_instytucji::text <> 'NFZ'::text THEN 'INNE'::text ELSE NULL::text END AS rodzaj\_faktury, faktura.faktura\_id, numerfaktury(faktura.\*) AS nr\_dokumentu, CASE WHEN faktura.kupujaca\_instytucja\_id IS NOT NULL THEN replace(instytucja.nip::text, '-'::text, *::text)*  ELSE 'OI'::text END AS id\_kontr, faktura.data\_dokumentu,  CASE WHEN faktura.termin\_platnosci IS NULL THEN faktura.data\_dokumentu ELSE faktura.termin\_platnosci END AS termin\_platnosci, "substring"(faktura.uwagi::text, 1, 255) AS uwagi FROM faktura LEFT JOIN instytucja ON instytucja.instytucja\_id = faktura.kupujaca\_instytucja\_id WHERE faktura.jednostka\_id IS NULL AND faktura.prefiks\_faktury IS NOT NULL AND (faktura.typ\_faktury::text = 'W'::text AND faktura.kupujaca\_instytucja\_id IS NULL OR faktura.typ\_faktury::text = 'F'::text AND faktura.kupujaca\_instytucja\_id IS NOT NULL AND instytucja.typ\_instytucji::text = 'NFZ'::text OR faktura.typ\_faktury::text = 'F'::text AND faktura.kupujaca\_instytucja\_id IS NOT NULL AND instytucja.typ\_instytucji::text <> 'NFZ'::text); |

Opis kolumn:

rodzaj\_faktury - NFZ/KOMERCJA/INNE, nr\_dokumentu - numer faktury, id\_kontr - NIP kupującego, data\_dokumentu,

termin\_płatności, uwagi - zanotowane uwagi dot. faktury.

Przykładowe wykorzystanie:

SELECT \* FROM simple\_faktura;

**Widok simple\_pozycja**

zwraca informacje dotyczące pozycji faktur sprzedaży Zapytanie:

|  |
| --- |
| CREATE OR REPLACE VIEW simple\_pozycja AS SELECT DISTINCT ON (pozycja\_faktury.pozycja\_faktury\_id) pozycja\_faktury.faktura\_id, pozycja\_faktury.pozycja\_faktury\_id, numerfaktury(faktura.\*) AS nr\_dokumentu, pozycja\_faktury.lp, jm.jm\_id AS indeks\_jm\_id, indeks.indeks\_kod,  CASE WHEN pozycja\_jednostka.typ\_jednostki::text = 'PZ'::text THEN pozycja\_jednostka.kod\_opk WHEN pozycja\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN pozycja\_nadrz\_jednostka.kod\_opk WHEN pozycja\_nadrz\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN pozycja\_nadrz\_nadrz\_jednostka.kod\_opk WHEN faktura\_jednostka.typ\_jednostki::text = 'PZ'::text THEN faktura\_jednostka.kod\_opk WHEN faktura\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN faktura\_nadrz\_jednostka.kod\_opk WHEN faktura\_nadrz\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN faktura\_nadrz\_nadrz\_jednostka.kod\_opk ELSE NULL::character varying END AS przychodnia\_opk,  CASE WHEN pozycja\_jednostka.typ\_jednostki::text = 'PZ'::text THEN pozycja\_jednostka.konto\_fk WHEN pozycja\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN pozycja\_nadrz\_jednostka.konto\_fk WHEN pozycja\_nadrz\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN pozycja\_nadrz\_nadrz\_jednostka.konto\_fk WHEN faktura\_jednostka.typ\_jednostki::text = 'PZ'::text THEN faktura\_jednostka.konto\_fk WHEN faktura\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN faktura\_nadrz\_jednostka.konto\_fk WHEN faktura\_nadrz\_nadrz\_jednostka.typ\_jednostki::text = 'PZ'::text THEN faktura\_nadrz\_nadrz\_jednostka.konto\_fk ELSE NULL::character varying END AS przychodnia\_konto,  CASE WHEN pozycja\_jednostka.kod\_opk IS NOT NULL THEN pozycja\_jednostka.kod\_opk |

|  |
| --- |
|  WHEN pozycja\_nadrz\_jednostka.typ\_jednostki::text = 'P'::text THEN pozycja\_nadrz\_jednostka.kod\_opk WHEN faktura\_jednostka.kod\_opk IS NOT NULL THEN faktura\_jednostka.kod\_opk WHEN faktura\_nadrz\_jednostka.typ\_jednostki::text = 'P'::text THEN faktura\_nadrz\_jednostka.kod\_opk ELSE NULL::character varying END AS poradnia\_opk,  CASE WHEN pozycja\_jednostka.konto\_fk IS NOT NULL THEN pozycja\_jednostka.konto\_fk WHEN pozycja\_nadrz\_jednostka.typ\_jednostki::text = 'P'::text THEN pozycja\_nadrz\_jednostka.konto\_fk WHEN faktura\_jednostka.konto\_fk IS NOT NULL THEN faktura\_jednostka.konto\_fk WHEN faktura\_nadrz\_jednostka.typ\_jednostki::text = 'P'::text THEN faktura\_nadrz\_jednostka.konto\_fk ELSE NULL::character varying END AS poradnia\_konto,  CASE WHEN umowa.umowa\_id IS NOT NULL THEN umowa.umowa\_kod\_fk WHEN umowa\_do\_faktury.umowa\_id IS NOT NULL THEN umowa\_do\_faktury.umowa\_kod\_fk ELSE NULL::character varying END AS kod\_umowy, pozycja\_faktury.ilosc, pozycja\_faktury.cena\_jedn\_netto, CASE WHEN pozycja\_faktury.stawka\_vat IS NULL THEN 'Pzw'::text ELSE 'P'::text || pozycja\_faktury.stawka\_vat::text END AS stawka\_vat, "substring"(pozycja\_faktury.uwagi::text, 1, 255) AS uwagi\_poz FROM pozycja\_faktury JOIN faktura ON faktura.faktura\_id = pozycja\_faktury.faktura\_id JOIN jm ON jm.jm\_id = pozycja\_faktury.jm\_id JOIN indeks ON indeks.indeks\_id = jm.indeks\_id LEFT JOIN jednostka pozycja\_jednostka ON pozycja\_jednostka.jednostka\_id = pozycja\_faktury.wyk\_jednostka\_id LEFT JOIN jednostka pozycja\_nadrz\_jednostka ON pozycja\_nadrz\_jednostka.jednostka\_id = pozycja\_jednostka.nadrz\_jednostka\_id LEFT JOIN jednostka pozycja\_nadrz\_nadrz\_jednostka ON pozycja\_nadrz\_nadrz\_jednostka.jednostka\_id = pozycja\_nadrz\_jednostka.nadrz\_jednostka\_id LEFT JOIN jednostka faktura\_jednostka ON faktura\_jednostka.jednostka\_id = faktura.wyk\_jednostka\_id LEFT JOIN jednostka faktura\_nadrz\_jednostka ON faktura\_nadrz\_jednostka.jednostka\_id =  |

faktura\_jednostka.nadrz\_jednostka\_id LEFT JOIN jednostka faktura\_nadrz\_nadrz\_jednostka ON faktura\_nadrz\_nadrz\_jednostka.jednostka\_id = faktura\_nadrz\_jednostka.nadrz\_jednostka\_id LEFT JOIN wyk\_zafakturowane ON wyk\_zafakturowane.pozycja\_faktury\_id = pozycja\_faktury.pozycja\_faktury\_id LEFT JOIN wykonanie ON wykonanie.wykonanie\_id = wyk\_zafakturowane.wykonanie\_id LEFT JOIN poz\_pak ON poz\_pak.poz\_pak\_id = wykonanie.poz\_pak\_id LEFT JOIN produkt\_jedn ON produkt\_jedn.produkt\_jedn\_id = poz\_pak.produkt\_jedn\_id LEFT JOIN umowa ON umowa.umowa\_id = produkt\_jedn.umowa\_id LEFT JOIN umowa umowa\_do\_faktury ON umowa\_do\_faktury.umowa\_id = faktura.umowa\_id WHERE faktura.typ\_faktury::text = 'F'::text AND faktura.jednostka\_id

IS NULL AND faktura.prefiks\_faktury IS NOT NULL

 ORDER BY pozycja\_faktury.pozycja\_faktury\_id;

Opis kolumn:

faktura\_id - identyfikator faktury, pozycja\_faktury\_id - identyfikator pozycji faktury, nr\_dokumentu - numer faktury, lp - liczba porządkowa pozycji na fakturze, indeks\_jm\_id - identyfikator jednostki miary (Indeksu towarów i usług), indeks\_kod - kod indeksu towarów i usług, przychodnia\_opk - kod MPK/OPK przychodni, przychodnia\_konto - fragment konta analitycznego przychodni, poradnia\_opk - kod MPK/OPK poradni, poradnia\_konto - fragment konta analitycznego poradni, kod\_umowy - kod umowy ilosc, cena\_jedn\_netto, stawka\_vat -stawka VAT (Pzw/P00/P03/P07/P08...), uwagi\_poz - zanotowane uwagi dot. pozycji.

Przykładowe wykorzystanie:

SELECT \* FROM simple\_pozycja;

### Magazyn

**Widok simple\_pozycja\_dok** zwraca informacje na temat pozycji dokumentów będącymi ruchami magazynowymi.

Zapytanie:

SELECT dokument.dokument\_id, dokument.typ\_dokumentu, dokument.data\_dokumentu, dokument.nr\_dokumentu, z\_jednostka.nazwa AS z\_komorka\_nazwa, z\_jednostka.kod\_opk AS z\_komorka\_kod\_opk, z\_jednostka.konto\_fk AS z\_komorka\_konto\_fk,

|  |
| --- |
| z\_instytucja.nazwa AS z\_instytucja\_nazwa, z\_instytucja.nip AS z\_instytucja\_nip, z\_instytucja.regon AS z\_instytucja\_regon, do\_jednostka.nazwa AS do\_komorka\_nazwa, do\_jednostka.kod\_opk AS do\_komorka\_kod\_opk, do\_jednostka.konto\_fk AS do\_komorka\_konto\_fk, do\_instytucja.nazwa AS do\_instytucja\_nazwa, do\_instytucja.nip AS do\_instytucja\_nip, do\_instytucja.regon AS do\_instytucja\_regon, pozycja\_dok.pozycja\_dok\_id, pozycja\_dok.lp, pozycja\_dok.wartosc, pozycja\_dok.typ\_dodatkowy, grupa.nazwa\_z\_hierarchia, grupa.grupa\_kod, do\_pacjent\_ident.pesel, do\_pacjent\_ident.nazwisko, do\_pacjent\_ident.imie, CASE WHEN nadrz\_kontakt.ks\_numer IS NOT NULL THEN (((nadrz\_kontakt.ks\_prefiks::text || '/'::text) || nadrz\_kontakt.ks\_numer) || '/'::text) || nadrz\_kontakt.ks\_rok ELSE NULL::text END AS ks\_glowna\_nr,  CASE WHEN dokument.typ\_dokumentu::text = 'KOR'::text THEN NULL::integer ELSE dokument\_kor.dokument\_id END AS korekta\_dokument\_id,  CASE WHEN dokument\_kor.\* IS NULL THEN NULL::boolean WHEN (NOT (dokument\_kor.dokument\_id IN ( SELECT dokument.kor\_dokument\_id FROM dokument WHERE dokument.zatwierdzony))) IS NULL THEN true ELSE false END AS aktualna,  CASE WHEN dokument\_kor.\* IS NULL THEN false ELSE true END AS korekta, dokument\_kor.data\_dokumentu AS data\_korekty, pozycja\_dok.kor\_pozycja\_dok\_id, CASE WHEN dokument\_kor.\* IS NULL THEN NULL::date ELSE dokument\_kor.data\_dokumentu END AS korekta\_data\_korekty FROM pozycja\_dok JOIN dokument ON pozycja\_dok.dokument\_id = dokument.dokument\_id LEFT JOIN dokument dokument\_kor ON dokument\_kor.dokument\_id = pozycja\_dok.kor\_dokument\_id LEFT JOIN pozycja\_dok pozycja\_dok\_kor ON pozycja\_dok\_kor.pozycja\_dok\_id = pozycja\_dok.kor\_pozycja\_dok\_id JOIN jm ON jm.jm\_id = pozycja\_dok.jm\_id JOIN indeks ON indeks.indeks\_id = jm.indeks\_id JOIN grupa ON grupa.grupa\_id = indeks.grupa\_id |
|  LEFT JOIN jednostka z\_jednostka ON z\_jednostka.jednostka\_id = dokument.z\_jednostka\_id LEFT JOIN instytucja z\_instytucja ON z\_instytucja.instytucja\_id = dokument.z\_instytucja\_id LEFT JOIN jednostka do\_jednostka ON do\_jednostka.jednostka\_id = dokument.do\_jednostka\_id LEFT JOIN instytucja do\_instytucja ON do\_instytucja.instytucja\_id = dokument.do\_instytucja\_id LEFT JOIN kontakt do\_kontakt ON dokument.do\_kontakt\_id = do\_kontakt.kontakt\_id LEFT JOIN kontakt nadrz\_kontakt ON do\_kontakt.nadrz\_kontakt\_id = nadrz\_kontakt.kontakt\_id LEFT JOIN pacjent\_ident do\_pacjent\_ident ON do\_kontakt.pacjent\_ident\_id = do\_pacjent\_ident.pacjent\_ident\_id WHERE (dokument.typ\_dokumentu::text <> ALL (ARRAY['ZM'::character varying::text, 'ZZ'::character varying::text, 'PAP'::character varying::text,  'WAP'::character varying::text, 'MMK'::character varying::text, 'PZK'::character varying::text, 'ZMK'::character varying::text,  'WZK'::character varying::text])) AND dokument.zatwierdzony ORDER BY pozycja\_dok.dokument\_id, pozycja\_dok.lp, pozycja\_dok.pozycja\_dok\_id; |

Opis kolumn:

dokument\_id - identyfikator dokumentu, typ\_dokumentu - typ dokumentu, data\_dokumentu - data dokumentu, nr\_dokumentu - numer dokumentu, z\_komorka\_nazwa - nazwa komórki wydającej, z\_komorka\_kod\_opk - kod OPK/MPK komórki wydającej, z\_komorka\_konto\_fk - fragment konta dla FK (fragment konta analitycznego) komórki wydającej, z\_instytucja\_nazwa - nazwa firmy zewnętrznej (dostawca), z\_instytucja\_nip - NIP firmy zewnętrznej (dostawca), z\_instytucja\_regon - regon firmy zewnętrznej (dostawca), do\_komorka\_nazwa - nazwa komórki przyjmującej, do\_komorka\_kod\_opk - kod OPK/MPK komórki przyjmującej, do\_komorka\_konto\_fk - fragment konta dla FK (fragment konta analitycznego) komórki przyjmującej, do\_instytucja\_nazwa - nazwa firmy zewnętrznej (odbiorca), do\_instytucja\_nip - NIP firmy zewnętrznej (odbiorca), do\_instytucja\_regon - regon firmy zewnętrznej (odbiorca), pozycja\_dok\_id - identyfikator pozycji dokumentu, lp - liczba porządkowa w dokumencie, wartość - wartość pozycji, typ\_dodatkowy - rodzaj ceny, nazwa\_z\_hierarchia - nazwa z hierarchią (grupy towarowej), grupa\_kod - kod grupy towarowej, pesel - pesel pacjenta,

nazwisko - nazwisko pacjenta, imie - imię pacjenta, ks\_glowna\_nr - numer księgi pacjenta, korekta\_dokument\_id - identyfikator dokumentu korygującego (informuje jaki dokument miał wpływ na powstanie tej pozycji), aktualna - uzupełniana jedynie dla korekt - informuje czy dana korekta jest aktualną (należy do aktualnej korekty - ostatnio dodanej), korekta - informuje czy pozycja jest korektą, data\_korekty - data korekty, kor\_pozycja\_dok\_id - identyfikator pozycji skorygowanej przez daną pozycję, korekta\_data\_korekty - data dokumentu, który spowodował korektę pozycji.

Przykładowe wykorzystanie: - wszystkie pozycje:

SELECT \* FROM simple\_pozycja\_dok;

* wszystkie pozycje aktualne (pomijanie nieaktualnych korekt):

SELECT \* FROM simple\_pozycja\_dok WHERE aktualna is null or aktualna = true;

* pozycje, które są korektami:

SELECT \* FROM simple\_pozycja\_dok WHERE korekta = true;

**Dokumenty magazynowe dostępne w widokach**

**Rodzaje widocznych dokumentów**

 BO - bilans otwarcia, dokument ten ma uzupełnione pola do\_komorka\_ - wskazanie na jednostkę wykonującą remanent, PZ - faktura/przyjęcie towaru - dokument ten ma uzupełnione pola z\_instytucj\_\* - dostawca leków oraz do\_komorka\_\* - apteka przyjmująca towar, KOR - korekty faktur - dokument ten ma uzupełnione pola z\_instytucj\_\*

* dostawca leków oraz do\_komorka\_\* - apteka przyjmująca towar,

 WZ - wydanie zewnętrzne. Rozróżniamy dwa rodzaje wydania zewnętrznego - do firmy zewnętrznej oraz do pacjenta. Zawsze w dokumencie wydania uzupełnione są pola z\_komorka\_\* - dane jednostki wydającej, w zależności od rodzaju uzupełniane są różne pola odbiorcy wydania, albo uzupełniane są dane firmy - pola do\_instytucja\_\* albo dane pacjenta (pesel, nazwisko, imie, ks\_glowna\_nr). MM - przekazanie między jednostkami, dokument ma uzupełnione zarówno pola z\_komorka\_\* jaki i do\_komorka\_\*. W widoku simple\_dokument każdy dokument MM występuje dwukrotnie - jako wydanie z jednostki oraz jako przyjęcie do jednostki, SN - strata nadzwyczajna - dokument ma uzupełnione pola z\_komorka\_\*, w przypadku tego dokumentu nie istnieje magazyn do którego jest skierowany, ZW - zużycie własne - dokument ma uzupełnione pola z\_komorka\_\*, w przypadku tego dokumentu nie istnieje magazyn do którego jest skierowany,

 PR - dokument produkcji, dla którego pierwsza pozycja (lp 1) stanowi produkowany lek, natomiast pozostałe pozycje stanowią składniki produkcji.

W widoku simple\_pozcja\_dok widoczne są tylko dokumenty zatwierdzone, natomiast w widoku simple\_dokument również widoczne są dokumenty niezatwierdzone.

**Pobranie danych korekt**

Poniżej znajduje się przykład prezentacji danych korekty pozycji dokumentu. Wszystkie zaprezentowane na zrzucie ekranu wpisy dotyczą jednej pozycji dokumentu o identyfikatorze 752878. Zrzut ekranu odnosi się do następującego przykładu:

Dnia 2013.01.01 zarejestrowano fakturę zawierającą pozycję w której było 10 opakowań leku w cenie 10 zł za opakowanie, 3 stycznia lek ten w liczbie 5 opakowań został przekazany dokumentem MM na oddział. Dnia 2013.02.01 została zarejestrowana korekta faktury, korygująca zarówno ilość jak i wartość. Przed korektą: 10 op. po 10 zł, po korekcie 9 op. po 20 zł za opakowanie. Dnia 2013.03.01 została zarejestrowana nowa korekta faktury, stan po korekcie 8 op. po 10zł.

**Ważna informacja**: Korygowane są zawsze faktury, nie występuje korygowanie korekt.



Opis zrzutu ekranu: wiersz pierwszy jest to pozycja dokumentu przyjęta fakturą 2013.01.01, wartość pozycji to 100 zł, wiersz drugi to przekazanie pozycji dokumentem MM do oddziału, wartość to 50zł (5op. po 10 złotych). Wiersz trzeci to pierwsza korekta faktury (pozycji faktury z wiersza 1) zwiększająca o 80 wartość faktury. Wiersz 4 jest konsekwencją korekty - jest to pozycja dokumentu dopisana do dokumentu MM, pole korekta\_dokument\_id wskazuje identyfikator faktury, kor\_pozycja\_dok\_id wskazuje na identyfikator korygowanej pozycji, korekta\_data\_korekty wskazuje datę korekty (pozycja korekty jest dopisana do dokumentu MM więc w polu data\_dokumentu znajduje się data dokumentu MM). Wartość w wierszu czwartym to 50 - o tyle zwiększyła się wartość pozycji po korekcie. Wiersz 5 to kolejna korekta tej samej pozycji, wiersz 6 to konsekwencja korekty. Dodatkowe flagi: korekta, aktualna są umieszczone w widoku w celu łatwiejszego wybierania danych.

### Rejestracja czasu pracy

Widoki zawierające zarejestrowany w systemie czas pracy personelu z możliwością podziału na klasy użytkowników, jednostki organizacyjne, rodzaje dyżurów. Widoki zawierają również informacje o zarejestrowanych absencjach personelu oraz godzinach opuszczenia miejsca pracy.

**Widok rcp\_dane** Opis kolumn:

rcp\_id - identyfikator rekordu jednostka\_id - identyfikator jednostki organizacyjnej jednostka\_kod\_opk - kod ośrodka powstawania kosztów jednostka\_nazwa - identyfikator rekordu user\_id - identyfikator rekordu użytkownika user\_kod\_fk - kod użytkownika dla zewnętrznego systemu F-K user\_lastname - nazwisko użytkownika user\_firstname - imię użytkownika user\_pesel - pesel użytkownika

user\_nr\_prawa\_wyk\_zaw - numer prawa wykonywania zawodu użytkownika klasa\_uzytkownika\_kod - kod klasy użytkownika (kod wewnętrzny np 1) klasa\_uzytkownika\_opis - opis kodu klasy użytkownika (np lekarz, pielęgniarka) dzien - data w formacie date czas\_nominalny\_z\_harmonogramu - czas pracy z harmonogramu w formacie time czas\_faktyczny\_przepracowany - czas faktycznie przepracowany w formacie time godziny\_nadliczbowe\_50 - godziny nadliczbowe 50% w formacie time godziny\_nadliczbowe\_100 - godziny nadliczbowe 100% w formacie time godz\_swiateczne\_45 - godziny świąteczne 45% godz\_nocne\_20 - godziny nocne 20% w formacie time godz\_nocne\_30 - godziny nocne 30% w formacie time godz\_nocne\_65 - godziny nocne 65% w formacie time godz\_pogotowia\_20 - godziny pogotowia 20% w formacie time godz\_pogotowia\_30 - godziny pogotowia 30% w formacie time dyzur\_lekarski\_zwykly\_50 - dyżur lekarski zwykły 50% w formacie time dyzur\_lekarski\_nocny\_100 - dyżur lekarski nocny 100% w formacie time dyzur\_lekarski\_swiateczny\_100 - dyżur lekarski świąteczny 100% w formacie time dyzur\_stazysci - dyżur stażyści w formacie time dyzur - dyżur w formacie time

Przykład pobrania danych - zapytanie sql, miesiąc styczeń 2012:

select \* from rcp\_dane where dzien >= '2012-01-01' and dzien <='2012-01-31';

**Widok rcp\_wyjscia** Opis kolumn:

rcp\_id - identyfikator rekordu rcp user\_id - identyfikator rekordu użytkownika user\_kod\_fk - kod użytkownika dla zewnętrznego systemu F-K dzien - data w formacie date wyjscie\_od - wyjście początek w formacie time wyjscie\_do - wyjście koniec w formacie time wyjscie\_kod - kod powodu wyjścja (kod wewnętrzny np 1) wyjscie\_opis - opis kodu powodu wyjścia

Przykład pobrania danych - zapytanie sql, wyjścia miesiąc styczeń 2012:

select \* from rcp\_wyjscia where wyjscie\_od is not null and wyjscie\_do is not null and dzien >= '2012-01-01' and dzien <='2012-01-31';

**Widok rcp\_absencja** Opis kolumn:

rcp\_id - identyfikator rekordu rcp user\_id - identyfikator rekordu użytkownika user\_kod\_fk - kod użytkownika dla zewnętrznego systemu F-K dzien - data w formacie date absencja\_kod - kod absencji (kod wewnętrzny np U) absencja\_opis - opis kodu abserncji (np urlop) absencja\_ilosc\_godzin - ilość godzin absencji w formacie time Przykład pobrania danych - zapytanie sql, absencje miesiąc styczeń 2012: select \* from rcp\_absencje where absencja\_ilosc\_godzin is not null and absencja\_kod is not null and dzien >= '2012-01-01' and dzien <='2012-01-31';