

Spis treści

1. System automatycznych zmian specyfikacji sprzedaży.....	1
2. Warunkowe blokowanie zapisu dokumentów.....	6
3. Kod SQL wywoływany dodatkowo przed zapisem dokumentu i po zapisie w danym magazynie.....	7

1. System automatycznych zmian specyfikacji sprzedaży.

System zaprojektowany został w celu umożliwienia zlecenia dowolnej firmie projektowania na przykład – promocji i dynamicznego ich zmieniania, bez ingerencji w kod programu MagAD.

W bazie programu MagAD znajduje się nierejestrowana w tablicy „tables_list” tablica:

Table "public.dane_skry_sql"			
Column	Type	Modifiers	Opis
uzyj	boolean		czy używać danej linii wpisu
symb	character varying(50)	not null	symbol wpisu (w jego ramach wykonywane będą rozkazy SQL w kolejności lp)
lp	double precision	not null	kolejność wykonania polecenia SQL zawartego w kolumnie „tresc”
tresc	text		polecenie SQL – teksty i symbole mogą być wymieniane przez program MagAD – są traktowane, jak pewna forma zmiennych i są podawane z poziomu procedury wywołującej zapytania z zakresu danego „symbolu”
warunek	text		polecenie SQL – jeżeli istnieje, to jest wykonywane przed zapytaniem z pola „tresc” i od jego wyniku zależy, czy zapytanie z pola „tresc” będzie w ogóle wykonywane (więcej szczegółów znajduje się w treści poniżej)

Indexes:

"dane_skry_sql_pkey" PRIMARY KEY, btree (symb, lp)

Promocje (SAZSS) wpisywane są jako komendy SQL do pola „tresc” ww. tablicy. Rozpoznawane są przez program, jako SAZSS poprzez wpisy w polu „symb”: promo1.sql, promo1_1.sql, promo1_2.sql, promo1_3.sql, promo1_4.sql. Jeżeli nie istnieje wpis „promo1.sql”, albo nie jest uaktywniony poprzez wpis TRUE w polu „uzyj”, pozostałe wpisy będą ignorowane przez program MagAD.

Komendy te nie mogą włączać, zatwierdzać czy odwoływać transakcji – nie wolno używać operacji w transakcjach.

Działanie wpisów w poszczególnych momentach pracy w programie:

* promo1.sql, promo1_1.sql, promo1_2 - wykonywane przy każdym dopisaniu towaru

* promo1_2, promo1_3 - wykonywane przy każdej edycji, zerowaniu ilości na specyfikacji, usunięciu pozycji, pobraniu danych z pliku roboczego, przy

wejściu do specyfikacji (jesli nie jest pusta) w opcji wprowadzania nowych, lub wprowadzania zmian na specyfikacji
* promo1_2, promo1_4 przy zapisie specyfikacji; zwrocenie przez "promo1_2" wynik ostatniego zapytania SELECT, jako pierwsza kolumna pierwszego wiersza wyniku tego SELECT
w wartosci "1" warunkuje uruchomienie promo1_4 .

Zasadniczo wpisy w polu „tresc”, mające ten sam wpis w polu „symb” można uznać po prostu za skrypt SQL, wykonywany w kolejności „lp”. „Lp” umyslnie jest liczba rzeczywista, aby nie było konieczności renumerowania kolejności poleceń jeśli chcemy wstawić jakiś wiersz. „Lp” może przybrać np. wartość 18.23.

Skrypty z pola „tresc” uruchamiane są tylko w momencie, kiedy wprowadzane są paragony, polecenia w kolejności „lp” w ramach jednego wpisu „symb”. Jeżeli w polu „warunek” znajduje się zapytanie, to sposób jego wykonania warunkuje uruchomienie zapytania z pola „tresc”. Jest ono wykonywane przed zapytaniem z pola „tresc”. Jeżeli zapytanie z pola „warunek” nie zwróci rekordów, albo zwróci, ale pierwszy rekord/pierwsze pole zwracanego wyniku nie zwróci 1, to zapytanie z pola „tresc” w ogóle nie zostanie wykonane.

Skrypty zwracają do programu:

- 1) wynik pierwszego zapytania SELECT, jako pierwsza kolumna pierwszego wiersza wyniku tego SELECT
- jest to tekst do 255 znaków - są to komentarze, przywoływane na ekran – jeżeli chcemy w tych komentarzach łamać linie, to zostanie ona złamana w miejscu, gdzie w tekście wystąpi znak „@”
- 2) wynik ostatniego zapytania SELECT, jako pierwsza kolumna pierwszego wiersza wyniku tego SELECT - jeśli wynikiem jest "1", to pokazywany jest komunikat, bazujący na wyniku pierwszego zapytania SELECT (pierwsza kolumna pierwszego wiersza wyniku tego SELECT) .

W każdej komendzie SQL (z pola „tresc” lub/i „warunek”) każdy napis SPECAKT_ zamieniany będzie na nazwę tablicy ze specyfikacją aktualną (towary wypisywane przez użytkownika), MAGAKT_ na tablicę stanów magazynu aktualnego.

W przypadku programu używającego synchronizacji danych z centralą funkcjonuje jeszcze tabela (nierejestrowana w „tables_list”, można ją założyć programem „upg34_35”, podając opcję 63):

dane_nowe_sazss (
indeks varchar(15), - indeks numeryczny towaru w czasie wgrywania przesunięcia
nazwa1 varchar(31), - pierwsza część nazwy towaru w czasie wgrywania przesunięcia
nazwa2 varchar(30), - pierwsza część nazwy towaru w czasie wgrywania przesunięcia
cena_br varchar(36), - cena brutto towaru w czasie wgrywania przesunięcia
opis varchar(250), - pole przeznaczone do opisu np. Rabatów obowiązujących dla pozycji
kod_sazss varchar(10), - kod rodzaju zmiany, który obowiązywał w czasie pobierania przesunięcia
nr_wewn varchar(10), - musi być to numer wewnętrzny pozycji w katalogu towarów, skonwertowany na tekst
mag_skl int, - numer magazynu w jednostce podległej, do którego wgrywane jest przesunięcie przychodu (typ = 18)
nr_przes int, - numer (zakodowany rok i numer) przesunięcia przychodu (typ = 18)
znak_przes varchar(4), - znak przesunięcia przychodu (typ = 18)

data_przes date, - data przesunięcia przysходу (typ = 18)

czas_zatw timestamp, zatw_uz_nr int – czas w którym operator zatwierdził wyskakująca na początku działania programu listę, pokazująca towary zmieniane (promocyjne)

, zatw_uz_naz varchar(55), - nazwa operatora, który zatwierdził dane

czas_importu timestamp default now() - czas zaimportowania przesunięcia

);

Tabela nie ma powiązanych żadnych indeksów i klucza głównego.

Służy ona do gromadzenia informacji o towarach docierających na przesunięciach, które spełniają aktualnie ustawioną promocję (czyli podlegają zmianom podczas wprowadzania sprzedaży). Przy starcie programu pokazywana jest zawartość tej tabeli dla „czas_zatw is NULL”, Pole „opis” jest tak długie, ponieważ jednej pozycji mogą towarzyszyć np. różne rabaty – można więc je w tym polu wyszczególnić wraz z opisami.

Zasadniczo pola od „indeks” do „opis” mogą być wypełniane dowolną informacją, dotyczącą towarów.

Przeglądanie tablicy po zatwierdzeniu danych jest możliwe w opcji analiz specjalnych.

Dane do tablicy dopisywane są automatycznie przy pobieraniu przesunięć do jednostki z centrali, o ile w tablicy „dane_skry_sql” umieszczone są do wykonania komendy dla wpisu symb=’ promo1_0.sql’ - patrz dalej opis wpisów sterujących w tabeli „dane_skry_sql” dla symbolu ‘ promo1_0.sql’.

Reszta wyjaśniona jest na przykładzie promocji (oczywiście SQL tej promocji może zawierać pewne niedociągnięcia, ale w praktyce promocja działała bez problemów):

Zasady promocji:

Na jednym paragonie :

Na paragonie 2 pozycje, które muszą być z 2 różnych grup (aby promocja była aktywna).

GRUPA 1 ZAWIERA (LUB):

BOTKI D, KOZAKI D, TRAPERY D

GRUPA 2 ZAWIERA (LUB):

SPORT DAMSKI, TEKSTYL DAMSKI, SANDAŁY D, POLBUTY D, MOKASYNY D, KŁAPKI D, KAPCIE D, KALOSZE D, CZOLENKA D, BALETKI D

Jeśli już mamy na paragonie 2 pozycje i jedna należy do jednej grupy a druga do drugiej, na tańszą z tych pozycji nalicza się rabat w wysokości 50%.

Nie ma komunikatów o niespełnieniu warunków promocji.

Jeśli spełniona jest promocja, jest komunikat, informujący też o tym, że

promocji nie będzie, jeśli info przestanie się pojawiać po dopisaniu..

Znaczenie wpisów sterujących w polu „symb” (skryptów, jak je nazwałem także wcześniej):

* **promo1.sql** - nie robi nic, ale musi być

* **promo1_1.sql** - sprawdza, czy dopisano towar podlegający promocji i zwraca wtedy "1" (ostatnie zapytanie SELECT skryptu) oraz zwraca komunikat, że dopisano, jeśli to prawda (komunikat - patrz - pierwsza linia skryptu promo1_1)

* **promo1_2.sql** - sprawdza (i wtedy ostatecznie zap.SELECT zwróci "1") i ostrzega,

że przy zapisie danych będzie wykonana przecena

(za pomocą pierwszego zapytania SELECT); informuje o sumie paragonu po

promocji;

jeśli wpisane ceny różnią się od tych ze stanu, bo ktoś zmienił, to i tak wymusza promocje wg zmian ceny od cen katalogowych, ponieważ skrypt ustawia też wszystkim towarom, podlegającym promocji ceny na katalogowe i zbija je, jeśli ceny były różne - de facto nie da nigdy zmienić ceny, powracając do katalogowej - pilnuje zachowania kolejności wpisania na specyfikację.

Jeśli ilość jakiejś pozycji o tym samym numerze wewn>10 to nie ma spełnionych warunków promocji;

Przecena robiona jest z cen z tablicy stanów, niezależnie od wpisanych cen. Docelową przecenę przed zapisem paragonu zrealizuje skrypt promo1_4.

W tablicy t umieszczane są warunki spełnienia promocji i ddo wyliczeń sum oraz OID pozycji z promocji.. Pole "a1" jest nadmiarowe i zawsze TRUE.

* **promo1_3.sql** - informuje, że na specyfikacji nie ma układu towarów spełniających warunki przeceny promocyjnej (jeśli sprawdzenie w promo1_2 wyjdzie na nie, czyli ostatni SELECT promo1_2 nie zwróci "1")

aktualnie komunikat ten jest wyłączony

* **promo1_4.sql** - wywoływany tylko po uruchomieniu promo1_2; wykonuje zmiany cen na specyfikacji przed zapisem, o ile ostatni SELECT promo1_2 zwróci "1"; zasady zachowania bardzo podobne do promo1_2, tylko promo1_2 bada potencjalne możliwości zmian, a promo1_4 je wprowadza.

* **promo1_0.sql** – nie musi występować. W systemach z synchronizowaniem danych pomiędzy oddziałami służy w procesie notowania przychodzących na przesunięciach do jednostki podległej towarów stworzenia tablicy tymczasowej „tmp_z_wl” ze wszystkimi towarami występującymi w promocji. Dane w tabeli „tmp_z_wl” muszą mieć taką strukturę, jak tablica „dane_nowe_sazss” do pola „nr_wewn”, czyli:

CREATE TABLE tmp_z_wl (indeks varchar(15),nazwa1 varchar(31),nazwa2 varchar(30),cena_br varchar(36),rabat varchar(250),kod_sazss

varchar(10),nr_wewn varchar(10)). Reakcją programu na takie wpisy w „dane_skry_sql” będzie uruchomienie obsługi tablicy „dane_nowe_sazss” (opisywana powyżej).

Zawartość skryptów z przykładu (polecenia są niekiedy długie, każde zakończone jest znakiem „;”, w tablicy „dane_skry_sql” opatrzone są numerami w kolumnie „lp” - tu są przedstawione bez tych numerów, we właściwej jednak kolejności:

Dla wpisu w „symb” = promo1.sql:

```
select '';
select 0;
```

Dla wpisu w „symb” = promo1_1.sql:

```
select 'Dopisano pozycję, której dotyczy aktualna promocja.@Nie zapomnij poinformowac o tym klienta.@Nie umieszczaj na paragonie więcej, niż po jednej@dwójce towarów spełniających promocję, nie udzielaj@samodzielnie rabatow.';
select 1 from SPECAKT_ as s where (upper(s.k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D','SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'))and(s.oid=(select max(oid) from SPECAKT_)) limit 1;
```

Dla wpisu w „symb” = promo1_2.sql:

```
create temporary table t(id int,a1 boolean,b1 boolean,c1 boolean,d1 boolean,suma float8,s1 float8,s2 float8);
insert into t values(1,TRUE,FALSE,FALSE,FALSE,0.0,0.0,0.0);
update t set b1=(select sum(ilosc) from SPECAKT_ where upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D','SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'))=2;
update t set c1=(select count(*) from SPECAKT_ where upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D'))=1;
update t set d1=(select count(*) from SPECAKT_ where upper(k2_wydawca) in ('SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'))=1;
update t set suma=(select sum(ilosc*cena_s_b) from SPECAKT_ where not upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D','SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'));
update t set s1=(select sum(s.ilosc*m.csb) from SPECAKT_ as s, MAGAKT_ as m where s.nr_wewn=m.nr_wewn and upper(s.k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D'));
update t set s2=(select sum(s.ilosc*m.csb) from SPECAKT_ as s, MAGAKT_ as m where s.nr_wewn=m.nr_wewn and upper(s.k2_wydawca) in ('SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'));
update t set suma=0 where suma is null;
update t set s1=0 where s1 is null;
update t set s2=0 where s2 is null;
update t set s1=s1/2 where s1<s2;
update t set s2=s2/2 where s2<=s1;
select 'Przy zapisie danych będzie wykonana przecena promocyjna.@Jezeli ten komunikat nie pojawi sie przy dopisaniu nastepnej@pozycji, przeceny nie bedzie.@Przewidywana wartość sprzedaży po przecenie = '||trim(trailing '0' from (cast ((select suma+s1+s2 from t limit 1) as varchar)));
select id from t where a1=TRUE and b1=TRUE and c1=TRUE and d1=TRUE;
drop table t;
```

Dla wpisu w „symb” = promo1_3.sql:

```
select 'Na specfikacji nie ma układu towarów spełniających warunki@przeceny promocyjnej.'
select 0;
```

Dla wpisu w „symb” = promo1_4.sql:

```
create temporary table t(id int,a1 boolean,b1 boolean,c1 boolean,d1 boolean,suma float8,s1 float8,s2 float8,nro1 bigint,nro2 bigint);
insert into t values(1,TRUE,FALSE,FALSE,FALSE,0.0,0.0,0.0,0.0,0.0);
update t set b1=(select sum(ilosc) from SPECAKT_ where upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D','SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'))=2;
update t set c1=(select count(*) from SPECAKT_ where upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D'))=1;
update t set d1=(select count(*) from SPECAKT_ where upper(k2_wydawca) in ('SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'))=1;
update t set suma=(select sum(ilosc*cena_s_b) from SPECAKT_ where not upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERY D','SPORT DAMSKI','TEKSTYL
```

```

DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'));
update t set s1=(select sum(s.ilosc*m.csb) from SPECAKT_ as s, MAGAKT_ as m where s.nr_wewn=m.nr_wewn and upper(s.k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERKY D'));
update t set s2=(select sum(s.ilosc*m.csb) from SPECAKT_ as s, MAGAKT_ as m where s.nr_wewn=m.nr_wewn and upper(s.k2_wydawca) in ('SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D'));
update t set suma=0 where suma is null;
update t set s1=0 where s1 is null;
update t set s2=0 where s2 is null;
update t set nro1=(select oid from SPECAKT_ where upper(k2_wydawca) in ('BOTKI D','KOZAKI D','TRAPERKY D') limit 1);
update t set nro2=(select oid from SPECAKT_ where upper(k2_wydawca) in ('SPORT DAMSKI','TEKSTYL DAMSKI','SANDALY D','POLBUTY D','MOKASYNY D','KLAPKI D','KAPCIED','KALOSZE D','CZOLENKA D','BALETKI D') limit 1);
update SPECAKT_ set cena_s_n=round(100*csn/2)/100,cena_s_b=round(100*csb/2)/100 from t,MAGAKT_ where SPECAKT_.oid=t.nro1 and SPECAKT_.nr_wewn=MAGAKT_.nr_wewn and s1<s2 and a1=TRUE and b1=TRUE and c1=TRUE and d1=TRUE;
update SPECAKT_ set cena_s_n=round(100*csn/2)/100,cena_s_b=round(100*csb/2)/100 from t,MAGAKT_ where SPECAKT_.oid=t.nro2 and SPECAKT_.nr_wewn=MAGAKT_.nr_wewn and s2<=s1 and a1=TRUE and b1=TRUE and c1=TRUE and d1=TRUE;
update SPECAKT_ set cena_s_n=csn,cena_s_b=csb from t,MAGAKT_ where SPECAKT_.oid=t.nro1 and SPECAKT_.nr_wewn=MAGAKT_.nr_wewn and s2<=s1 and a1=TRUE and b1=TRUE and c1=TRUE and d1=TRUE;
update SPECAKT_ set cena_s_n=csn,cena_s_b=csb from t,MAGAKT_ where SPECAKT_.oid=t.nro2 and SPECAKT_.nr_wewn=MAGAKT_.nr_wewn and s1<s2 and a1=TRUE and b1=TRUE and c1=TRUE and d1=TRUE;
drop table t;

```

2. Warunkowe blokowanie zapisu dokumentów.

Funkcjonalność ta pozwala na dosyć płynne, zależne od olbrzymiej ilości kombinacji czynników ograniczanie możliwości zapisu specyfikacji dokumentów towarowych.

Ograniczenie to dotyczy wszystkich użytkowników, którzy nie mają uprawnień maksymalnych, albo wszystkich poza Kierownikiem Systemu – w zależności od skonfigurowania.

Ograniczenie zakładamy w tablicy „dane_skry_sql”.

W polu symb ograniczenia takie mają wpis „ogr_spec_dokNR”, gdzie NR jest numerem typu dokumentu towarowego, dla którego jest ograniczenie (czyli liczby od 1 do 18).

Jeżeli w polu „uzyj” umieścimy true, to ograniczenie będzie włączone.

Ograniczenie może mieć wiele linii, które wykonywane są w kolejności liczb, wpisywanych do pola „lp”, wszystkie są zapytaniami niezwracającymi wyników (czyli np. insert, update, create itp.), natomiast dwie z linii z linii muszą zaczynać się od „select” i być zapytaniem zwracającym jedną kolumnę. Pierwsze zwraca „moc uprawnień” (cyfra 0,1). Drugie zawiera zwracany przez program MagAD komunikat. Nie może on mieć więcej, niż 150 znaków, łamanie linii komunikatu na ekranie następuje na znaku @. Jeżeli to pierwsze zaopytanie SELECT zwróci NULL lub pusty ciąg znaków (ale musi zwracać rekord, a nie być puste), oznacza to, że ograniczenie nie przeszkadza w zapisie dokumentu.

Moc uprawnień <>1, czyli np. 0 oznacza, że Kierownik Systemu otrzyma ostrzeżenie, reszta ma blokadę zapisu.

Moc uprawnień =1 oznacza, że operator z uprawnieniami maksymalnymi, w tym Kierownik Systemu otrzyma ostrzeżenie, reszta ma blokadę zapisu.

Jeżeli pierwsze zapytanie SELECT zwróci 0 lub 1 program reaguje pokazaniem komunikatu z drugiego zapytania SELECT i blokuje zapis dla nieuprawnionych użytkowników, a uprawnieni dostają takie ostrzeżenie plus komunikat, że pomimo to mogą zapisać i dopiero, jeśli nie chcą zapisać, program z zapisu zrezygnuje.

W w kolumnie „uzytk” przyporządkowuje się określone działanie tylko do użytkownika o wskazanym numerze. Jeśli jest -1, to działa dla wszystkich użytkowników.

Przykładowy wpis do tablicy „dane_skry_sql”, analizujący, czy można zapisać specyfikację paragonu (typ dokumentu=1) w zależności od tego, czy ilość > 5:

uzyj	symb	lp	tresc	symb_zm_wpr	uzytk	warunek
t	ogr_spec_dok1	1	create temporary table t(x int,czy boolean)		-1	
t	ogr_spec_dok1	2	insert into t values(0,null)		-1	
t	ogr_spec_dok1	3	update t set czy=(select sum(ilosc)>5 from SPECAKT_)		-1	
t	ogr_spec_dok1	4	select max(x) from t where czy=true		-1	
t	ogr_spec_dok1	5	select 'Ilość na specyfikacji może być <=5.'		-1	
t	ogr_spec_dok1	6	drop table t		-1	

Może być używana symbolika: SPECAKT_ oznacza tablicę aktualnie wprowadzanej specyfikacji dokumentu, MAGAKT_ tablicę stanów aktualnego magazynu (stany_mg), podobnie, jak przy ograniczeniach SAZSS oraz SPECMAG_ oznacza tablicę ze specyfikacjami dokumentów towarowych aktualnego magazynu, NRMAG_ numer aktualnego magazynu.

Podobnie, jak w wypadku SAZSS działa zawartość pola „warunek”.

W podawanym przykładzie zasadniczo linia dla lp=3 zawiera zapytanie sprawdzające. Natomiast linia dla lp=4 zawiera selekcję maksimum, ponieważ takie zapytanie musi zwrócić 0, albo NULL, ale nigdy zwracana ilość rekordów nie może być =0, ponieważ wynik takiego zapytania zostanie pominięty – program zacznie wtedy zachowywać się dziwnie, ponieważ nie zwróci w wyniku pierwszego napotkanego zapytania, zwracającego niepusty zestaw rekordów jako 0 lub 1, ale tekst zwracany przez SELECT z linii 5.

Powyższy schemat można stosować właściwie do każdej okazji, umiejętnie manewrując zawartością dla lp=3 i lp=5.

Należy zwrócić uwagę na to, skrypt może być wykorzystany do wykonania dowolnej operacji PRZED zapisem dokumentu, niekoniecznie musi blokować zapis dokumentu.

3. Kod SQL wywoływany dodatkowo przed zapisem dokumentu i po zapisie w danym magazynie.

Działa, jak warunkowe zapisywanie wybranych rodzajów dokumentów, ale nie daje żadnej reakcji na ekranie, ani nie prowadzi dialogu z operatorem. W polu „symb” identyfikowany jest poprzez wpis: „zapis_spec_dokX”, gdzie X to numer typu dokumentu. Skrypt ma analogiczne parametry, jak skrypt stosowany przy „warunkowym blokowaniu zapisu dokumentów”, ale dodany jest symbol „DODAJ_”, który może przyjmować wartości: 0-zdjęcie ze stanu, 1-dodanie do stanu, 2-zmiana ilości zamówionej w stanie, 3-nie zmienia stanu.

Jeżeli dokument to przesunięcie, próba wywołania takiego kodu wykonywana jest najpierw w jednym a potem drugim magazynie przesunięcia, dla typów 15 i 18.

Po zapisie działa analogiczny skrypt, identyfikowany w po lu „symb” poprzez wpis: „po_zapis_spec_dokX”, gdzie X to numer typu dokumentu.

Kolejne dodatkowe zmienne, dostępne już tylko w skryptach „po_zapis_spec_dokX”:

LISTA_DOK_MINUS_ → lista typów dokumentów dzielona przecinkami, które odejmują od stanu – jeśli jest pusta oznacza to, że żaden dokument nie zdajmuje ze stanu (na przykład wszystkie mają wyłączone zdejmowanie w konfiguracji),

LISTA_DOK_PLUS_ → lista typów dokumentów dzielona przecinkami, które dodają do stanu - jeśli jest pusta oznacza to, że żaden dokument nie dodaje do

stanu (na przykład wszystkie mają wyłączone dodawanie w konfiguracji),

CYZAM_ → wielkość logiczna – jeśli „true”, to zamówienia skonfigurowane są w MagAD tak, że rezerwują towar, jeśli „false”, to zamówienia nie rezerwują towaru.

Należy mieć też na uwadze, tworząc skrypty „po_zapis_spec_dokX”, że dostępna jest też dla nich tabela tymczasowa TMP_STANY_MG (widoczna tylko dla wywołującej skrypt sesji MagAD). Tabela ta zawiera m.in. w polu „nr_wewn” numery wewnętrzne (bez powtórzeń, numer wewnętrzny jest kluczem głównym w tej tabeli) towarów, które były przed zapisem i po zapisie dokumentu. Oferowana funkcja, która ustawia zarówno stany, jak i ilości zamówione „stan_ustaw_z_obrotow_wyb_lista_dok4” korzysta z tej tabeli.

Definicja funkcji „ stan_ustaw_z_obrotow_wyb_lista_dok4”

function stan_ustaw_z_obrotow_wyb_lista_dok4(specmag varchar,magakt varchar,lista_dok_minus varchar,lista_dok_plus varchar,czyzam boolean,cyuzupstan boolean) returns bigint

Wywołanie tej funkcji w skrypcie „po_zapis_spec_dokX” jest takie (sam zapis komend w kolumnie „tresc” dla każdego rodzaju dokumentu):

create temporary table t1(i bigint);

insert into t1 select stan_ustaw_z_obrotow_wyb_lista_dok4(SPECMAG_,MAGAKT_,LISTA_DOK_MINUS_,LISTA_DOK_PLUS_,CYZAM_,false);

drop table t1;