

*Dokument dotyczy wersji 3.4 i wyższych.*

### **Tworzenie własnych zestawień i funkcji własnych:**

W programie możliwe jest tworzenie własnych zestawień.

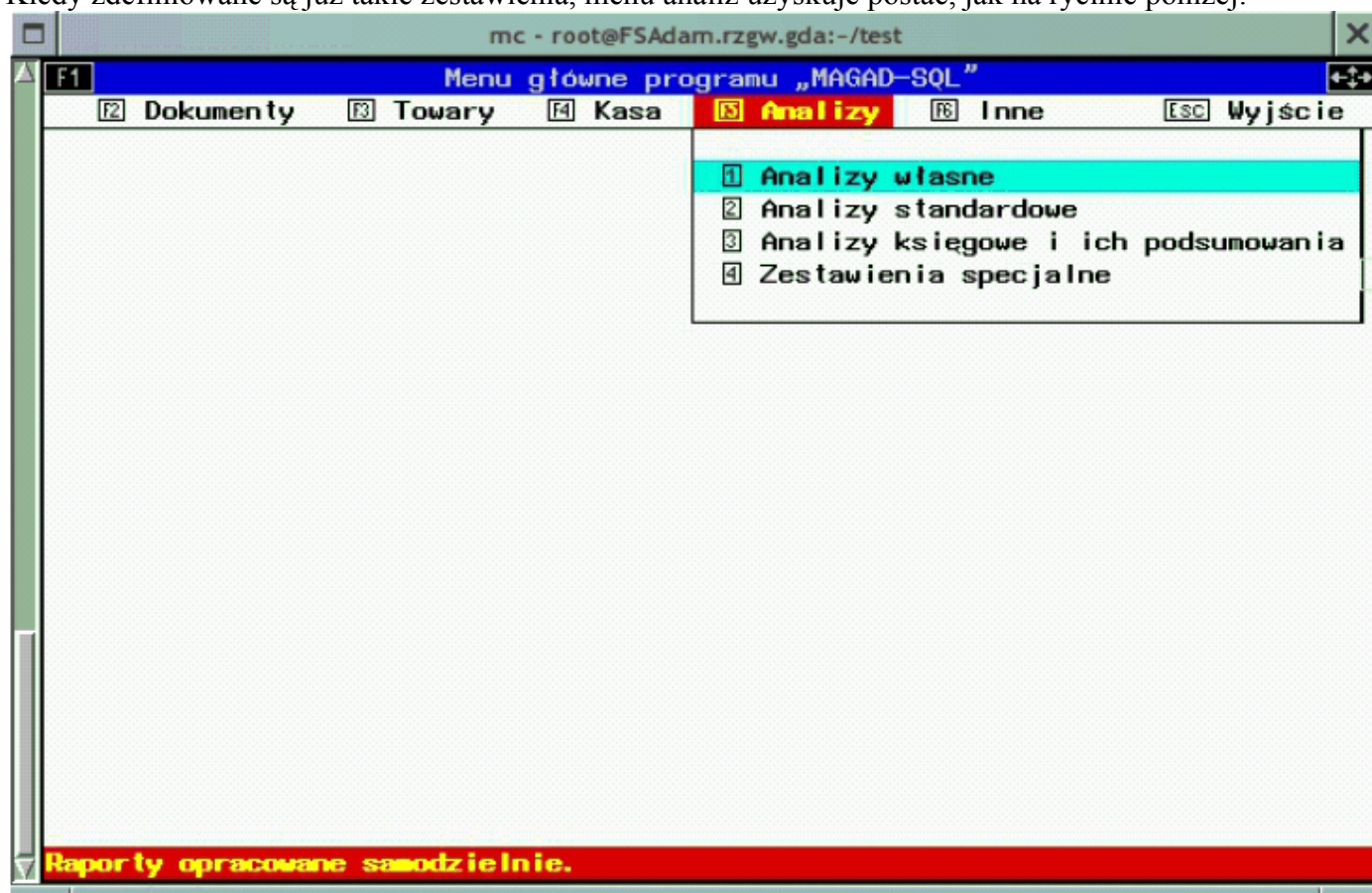
Od wersji 3.5 możliwe jest też na tej samej bazie tworzenie własnych dodatkowych funkcjonalności w programie – w rzeczywistości nie różni się to niczym od tworzenia zestawień własnych poza tym, że tablica, w której definiuje się takie funkcje własne, nazywa się „dane\_menu\_zsq”.

Tablica nie istnieje po wykonaniu standardowej instalacji – należy jednorazowo wejść do opcji „Inne”/”Funkcje własne i specjalne”/”funkcje własne” - tablica założy się automatycznie i od tej pory można jej używać.

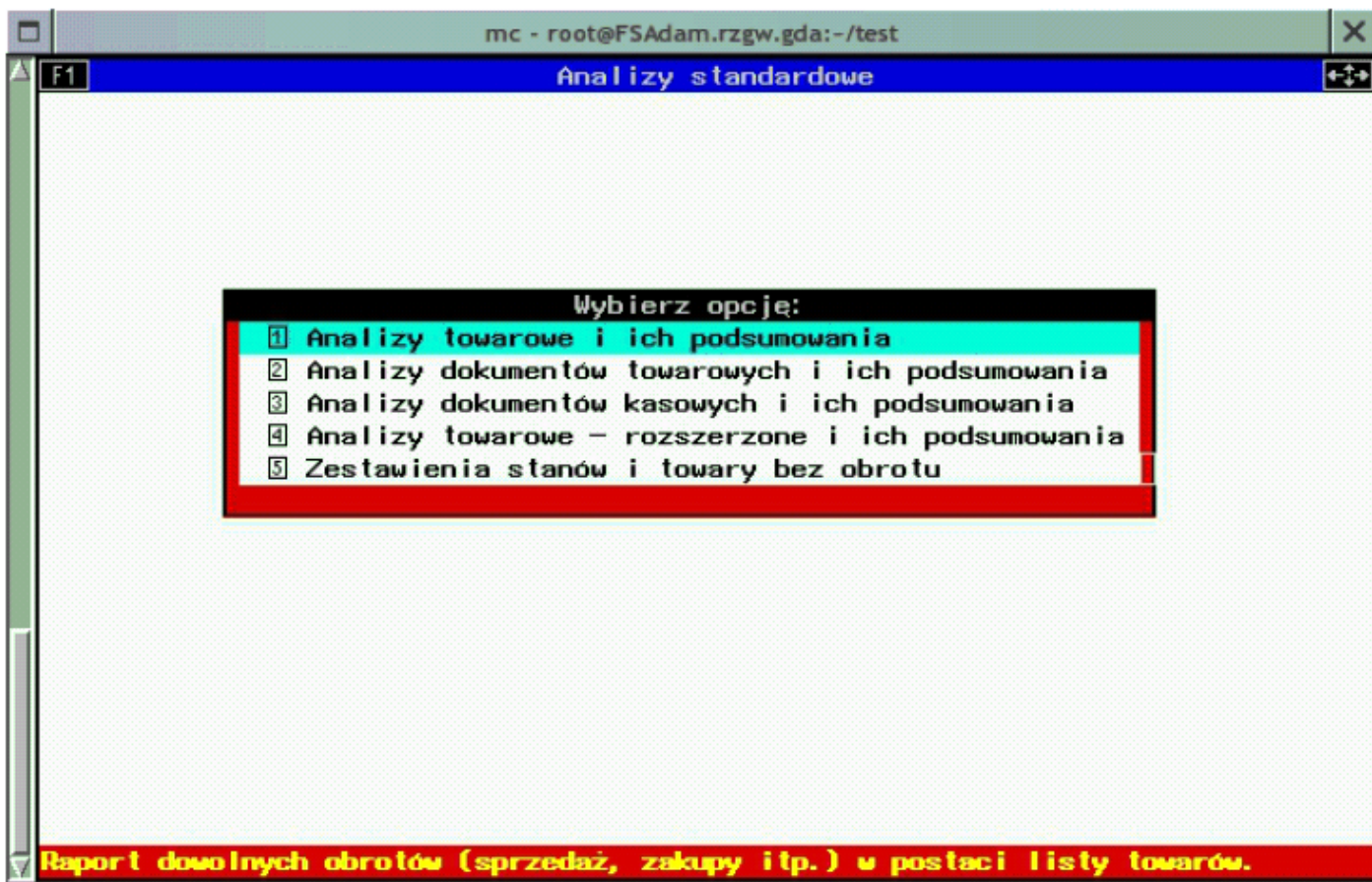
Jest to możliwe, o ile posiadamy znajomość języka SQL (język operacji na bazach danych). W praktyce pomocna taka funkcjonalność jest o tyle, że prosząc o pomoc dowolnego informatyka, znającego język SQL, można zdefiniować w programie do 9200 raportów, bez konieczności zmiany kodu programu.

Dodatkowo raporty te mogą być dla wybranych użytkowników inne, więc każdy może mieć dostęp do innych raportów.

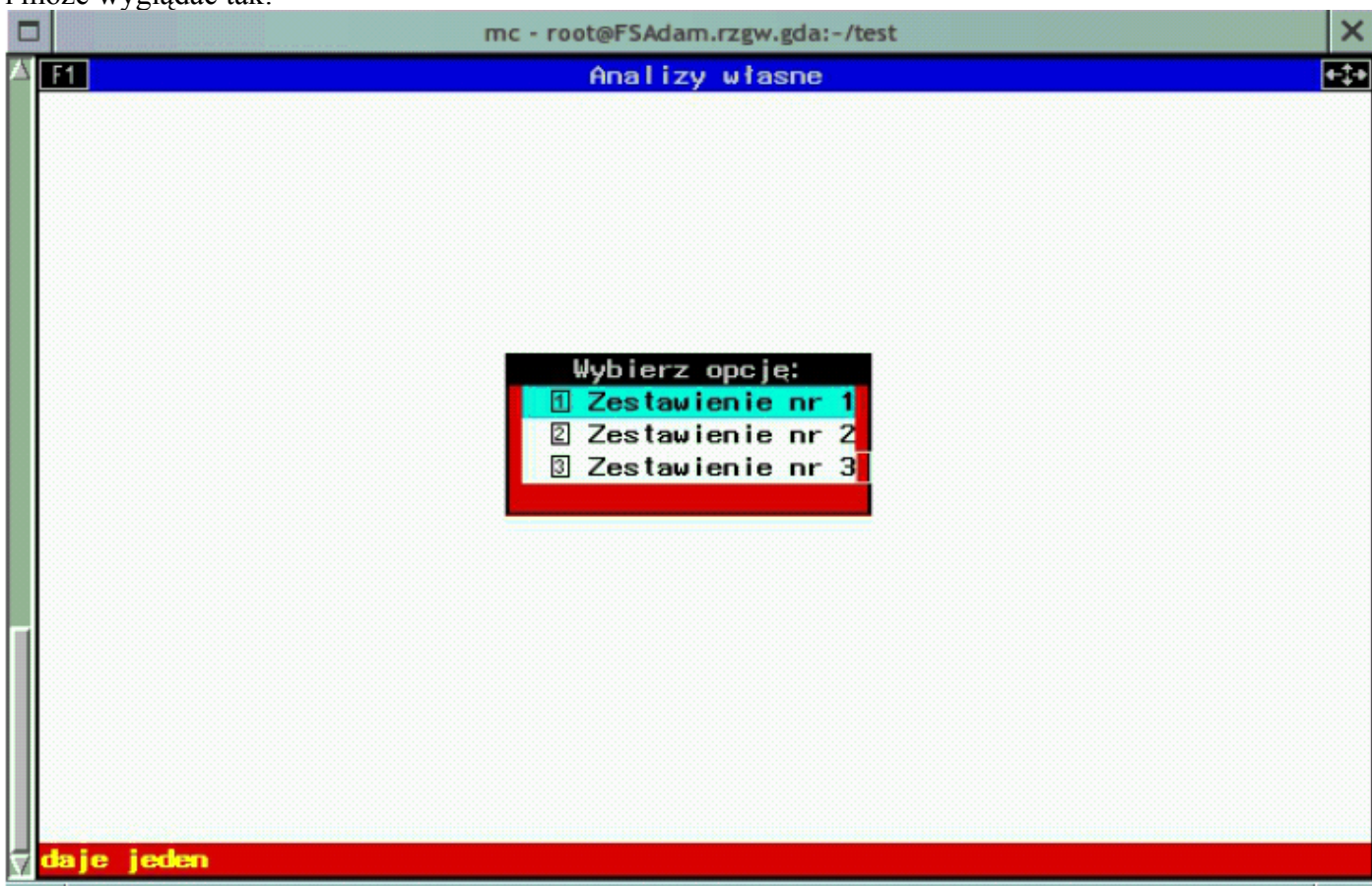
Kiedy zdefiniowane są już takie zestawienia, menu analiz uzyskuje postać, jak na rycinie poniżej:



Jak widać, „analizy księgowe i ich podsumowania” oraz „zestawienia specjalne” pozostaną w takim wypadku dostępne z menu. Reszta dostępnych zwykle zestawień „ukryta jest” w pozycji „analizy standardowe” - patrz rycina poniżej:



Jeżeli chodzi o menu „analiz własnych”, to jego zawartość i kolejne podmenu, widoczne nazwy opcji oraz podpowiedzi i pytania o ograniczenia i filtry dla raportów zależą od osoby, która takie zestawienia stworzy i może wyglądać tak:



Każdy raport można drukować, eksportować do Postscript/PDF, plików tekstowych, DBF, CSV.. Zestawienia „zadają” standardowe pytania, podobne do pytań z zestawień standardowych, ale tworzący może ograniczyć je tylko do takich, które uzgodni z firmą, dla której takie zestawienia zostaną wykonane.

Dalsze opisy (kursywą) są przeznaczone już dla osoby tworzącej takie raporty.

*Pierwsze i najważniejsze – to to, że tabela wynikowa raportu (jej nazwa – patrz dalej) nie może zawierać pól o większej długości, niż 250 (czyli de facto dotyczy to pól tekstowych) i tak należy ją zaprojektować. Jeśli jakiś opis jest dłuższy, należy w tabeli wynikowej podzielić go na dwa, czy więcej pól.*

*Definicje raportów zapisywane są w tabeli:*

Table "public.dane_menu_zes"		
Column	Type	Modifiers
uzytk	integer	not null
typ	smallint	not null
menu0	smallint	not null
menu1	smallint	not null
menu2	smallint	not null
linia	double precision	not null
nazwa	character varying(78)	not null
opis	character varying(78)	not null
typ_par	character varying(20)	not null
tresc	text	domyślnie ''::text
warunek	text	domyślnie ''::text (od wersji 3.5)

I

Indexes:

"dane\_menu\_zes\_pkey" PRIMARY KEY, btree (uzytk, typ, menu0, menu1, menu2, linia, nazwa, opis, typ\_par)

"dane\_menu\_zes\_oidindeks" UNIQUE, btree (oid)

"dane\_menu\_zes\_linia" btree (linia)

*Jeżeli uzytk=0 to zestawienie dostępne jest dla wszystkich użytkowników. Od wersji programu 3.5 zmieniono te zasady i dla wszystkich użytkowników parametr „uzytk”=-1, aby Kierownik Systemu też mógł mieć własny pakiet zestawień. Należy pamiętać, że numerowanie pozycji menu nie może się powtórzyć w zakresie zestawień jednego użytkownika, a więc należy uwzględnić w tej numeracji także zestawienia dostępne dla wszystkich.*

*Tabela ta musi być założona za pomocą programu do upgrade baz, aby we właściwy sposób została zarejestrowana do obsługi w MagAD. Zawartość tabeli można edytować za pomocą dowolnego narzędzia, w którym jest możliwa edycja tabel PostgreSQL (np. psql, PgAdmin, OpenOffice, MS Access).*

*Zapytania z numerem uzytk=0 widoczne są dla wszystkich użytkowników. Jeżeli uzytk=XXX, to jeżeli pracujemy, jako użytkownik nr XXX, to będziemy widzieć tylko określone dla nas raporty. Stąd – raportów może być 9200, ale dla każdego użytkownika. Kierownik Systemu widzi zestawienia z numerem uzytk=0.*

#### **Widoczne menu:**

*Definicje tekstów i odpowiedzi pozycji menu mają w kolumnie typ wartość 0.*

*Następnie sprawdzana jest zawartość kolumn menu0, menu1, menu2.*

*Jeżeli menu0=1, a menu1=0 i menu2=0 to mamy do czynienia z pierwszą pozycją głównego menu.*

*Jeżeli menu0=2, a menu1=0 i menu2=0 to mamy do czynienia z drugą pozycją głównego menu.*

*Jeżeli menu0=1, a menu1=1 i menu2=0 to mamy do czynienia z pierwszą pozycją pierwszego podmenu pierwszej pozycji głównego menu.*

*Jeżeli menu0=1, a menu1=2 i menu2=0 to mamy do czynienia z drugą pozycją pierwszego podmenu pierwszej pozycji głównego menu.*

*W kolumnie „nazwa” wpisany jest tekst widziany, jako pozycja w menu. W kolumnie „opis” odpowiedź, która widać w dolnej linijce programu MagAD (w odpowiedzi może być np. krótki, hasłowy opis raportu)*

#### **Zapytania tworzące raporty:**

*Jeżeli dla danej pozycji menu zdefiniowany jest raport, to wpisy w polach menu0, menu1, menu2 wskazują wyraźnie, do której pozycji menu przyporządkowane są jego elementy, czyli parametry, które podajemy i komendy SQL, które będą wykonywane. Parametry w polu "typ" mają wpisane 1, zapytania 2.*

*Żadne z zapytań nie może zwracać rekordów inaczej, niż zapisując je do tabeli.*

Raporty ładowane są do tablicy tymczasowej TMP\_Z\_WL którą zapytania muszą założyć w jakimś momencie. Zapytanie, które ogląda operator na ekranie jako raport to `SELECT * FROM TMP_Z_WL ORDER BY XXXXX`, gdzie XXXXX podajemy w polu "tresc", a pole "typ" ustawiamy na 3 dla danej pozycji menu. Pamiętajmy, że tablica TMP\_Z\_WL musi być założona, jako tymczasowa, ponieważ w przeciwnym wypadku użytkownicy w sieci będą sobie nawzajem przeszkadzać, robiąc raporty. To samo tyczy się wszystkich innych tablic, które są zakładane i usuwane w trakcie robienia raportu.

Po obejrzeniu wyniku program usuwa TMP\_Z\_WL. Kiedy wynik jest oglądany, może być wydrukowany, wydrukowany do pliku Postscript/PDF, zapisany w formacie DBF, CSV. W wypadku CSV dzielnikiem danych jest średnik, a teksty oczyszczone są ze znaków cudzysłowów i ujęte w cudzysłowy. Warto nadmienić, że dzielnikiem dziesiętnym w CSV generowanym z MagAD jest kropka, czyli w danych CSV będą liczby typu: „456.56”, a nie „456,56”.

### **Ustawianie parametrów, o które zapyta się program przed wykonaniem raportu:**

Należy przede wszystkim wspomnieć, że program pyta parametry raportu w kolejności wg danej w polu „linia”. Jeżeli w polu „linia” jest 0, program nie zapyta o taki parametr.

Jak wspomniałem, parametry w polu „typ” mają wpisana wartość 1.

**UWAGA !** Nazwa parametru nie może zawierać w sobie innej nazwy. Na przykład NIE WOLNO zastosować np. parametru „data\_od” i „pz\_data\_od” w jednej analizie.

### **Parametry są zasadniczo trzech rodzajów.**

**Pierwszy** rodzaj to stałe, które podajemy. Pytanie o taką stałą znajduje się w polu „opis”. Jej unikalny symbol – w polu „nazwa”. Symbol ten wykorzystujemy potem pisząc zapytania, tworząc raport – kiedy zapytanie będzie realizowane, nasz symbol zostanie zastąpiony odpowiednią stałą. Należy w zapytaniach zadbać o to, aby nasz symbol, jeśli symbolizuje ciąg znakowy, wziąć w „pojedyncze uszy”, np. 'dost\_ciaq', gdzie dost\_ciaq będzie nazwą parametru.

Sposób edycji będzie zależał od tekstu, który podamy, jako dana w polu „typ\_par”. Jeżeli dana w polu „typ\_par” ma więcej, niż jeden znak, program uzna, że mamy do czynienia ze wpisywaną wartością jakiejś stałej, czyli parametru pierwszego rodzaju.

W polu „typ\_par” dla parametrów pierwszego rodzaju mogą znaleźć się następujące wpisy, które spowodują możliwość wpisywania danych o określonej zawartości:

01 – wszystkie znaki

02 - data `**.**.****`, `**.**.**` wymagane wciskanie kropki

03 - cyfry liczb naturalnych i zero

04 - cyfry liczb rzeczywistych do 2 miejsc po przecinku

05 - nazwy plików (spacje i większość znaków niealfanumerycznych oraz narodowe są niedopuszczalne)

06 - cyfry liczb rzeczywistych do N miejsc po przecinku

07 - data `**_**_****`, `**_**_**` z automatycznym przeskokiem "-", nie pracują strzałki w bok, Home i delete; jeśli początkowa długość jest = 5, zamiast kresek jest ":"

08 - wszystkie, ale litery tylko wielkie

09 - cyfry liczb całkowitych

010 - cyfry liczb naturalnych i zero plus przecinek

011 - cyfry liczb rzeczywistych do N liczb po przecinku

012 - NIP `*****_***_**-**` z automatycznym przeskokiem "-", nie pracują strzałki w bok, Home i delete

013 - liczby rzeczywiste, znak -, +, . - jako wynik edycji tworzy sumę z wpisanych liczb

014 - jak 4, ale tylko dodatnie

015 - wszystkie, ale wyświetla gwiazdki

016 - cyfry liczb naturalnych i litery A..Z

Przy zastępowaniu parametru wpisana wartością stałe tekstowe poddawane są oczywiście zabiegowi konwertowania znaków, które takiej konwersji muszą być poddane do postaci akceptowalnej w zapytaniach. W polu „tresc” umieszczamy dane (są to stałe), która mają być proponowane, jako wstępna wartość parametru.

Od wersji 3.6.x: Jeżeli w polu „treść” umieścimy zapytanie „select ...”, i zwróci ono więcej, niż jeden wiersz, to jego wynik (pierwsza kolumna) zawartość stanie się lista wyboru, która uruchamia się po wejściu na pole edycji danej. Jeśli zwrócony będzie tylko jeden wiersz, nie będzie pokazywana lista wyboru. Dla dat trzeba, niestety, sformatować przekazanie przez taki SELECT daty. SQL zwraca datę w formacie RRRR-MM-DD, a MagAD w tym miejscu wymaga DD-MM-YY. Można to zrobić na różne sposoby, np. umieszczając w jednej linii oczywiście:

```
SELECT SUBSTRING(CAST(CURRENT_DATE AS VARCHAR(10)) FROM 9 for 2)||'-'||  
SUBSTRING(CAST(CURRENT_DATE AS VARCHAR(10)) FROM 6 for 2)||'-'||  
SUBSTRING(CAST(CURRENT_DATE AS VARCHAR(10)) FROM 3 for 2)
```

Zapytanie takie zwróci dzisiejszą datę w formacie DD-MM-RR.

Komenda zapytania nie może mieć więcej, niż 255 znaków. Jeśli w zapytaniu umieścimy tekst „TABW\_” to jest to odniesienie się do tabeli **TMP\_WYB**, wypełnianej w ramach **drugiego** (patrz niżej) typu danych filtrujących. Jeśli więc chcemy, aby nasze zapytanie odniosło się do wcześniej np. wybranych magazynów z listy, należy opcję wybierania magazynów umieścić wcześniej w definicji zestawienia własnego. Nie można odnosić się bezpośrednio do tabeli TMP\_WYB, ponieważ w momencie wpisywania ograniczeń analizy tabela ta nie istnieje jeszcze, ale jest jej wstępna kopia, której nazwę „zna” program MagAD i przekazuje na tej „zmiennej” „TABW\_” do zapytania. Należy mieć na uwadze, że nazwa tabeli „TABW\_” może mieć do 35 znaków i uwzględnić to w szacowaniu długości komendy zapytania. Po wybraniu wartości podpowiada się wybrana wartość z wiersza i pierwszej kolumny i nie jest ona obowiązkowa – operator może ją zmienić. Dodatkowo akceptowane są w omawianym zapytaniu „select” stałe UZYTEKAKTUALNY i MAGAKTUALNY (patrz opis trzeciego typu parametrów).

**Drugi** typ parametrów to listy wyboru „ptaszkowane”.

Musza one w polu „nazwa” mieć też swoje unikalne nazwy, ale dany typ dla danego raportu może wystąpić tylko raz, czyli dla takich parametrów dla jednego raportu w polu „typ\_par” nie mogą powtórzyć się wpisy. Jak widać np. informacja na temat może dla niektórych danych być powtórzona, ponieważ ma dwa lub więcej dopuszczalnych symboli (np. dostawcy, magazyny).

W polu „typ\_par” mogą być dla takich parametrów wpisy (zawsze jednoznakowe):

D - dostawcy

Z - dostawcy

k - dokumenty płatności i paragony gotówkowe

K - dokumenty płatności

P - dokumenty towarowe

U - użytkownicy (operatorzy z MagAD)

O - odbiorcy

m - magazyny

M - magazyny

F - magazyny

S – dana opisana w pliku konfiguracyjnym lista\_i2.txt w drugiej linii, najczęściej „gr.gł.” lub „SWW”

V - VAT

G - dana opisana w pliku konfiguracyjnym lista\_in.txt w czwartej linii, najczęściej „grupa towarowa”

9 - dana opisana w pliku konfiguracyjnym lista\_i2.txt w pierwszej linii, najczęściej „miejsce w magazynie”

7 - dana opisana w pliku konfiguracyjnym lista\_i2.txt w czwartej linii

8 - dana opisana w pliku konfiguracyjnym lista\_i2.txt w piątej linii

1,2,3,5 - dana opisana w pliku konfiguracyjnym lista\_in.txt w liniach numer (odpowiednio 1,2,3,5 – czwarta linia ma symbol G /patrz wyżej/)

Y – forma płatności

H - stan wyjaśniania dokumentu (nie dostępne w standardowej wersji programu)

A,B,C – (od wersji 3.5) dodatkowe pola z nagłówek dokumentów, odpowiednio nag1, nag2, nag3

I - (od wersji 3.5) dodatkowe pole nag4 z nagłówek dokumentów

J - (od wersji 3.5) dodatkowe pole nag5 z nagłówek dokumentów

L - (od wersji 3.5) dana z pola „cecha\_1” dla kontrahentów (odbiorców, dostawców, magazynów)

N - (od wersji 3.5) dana z pola „cecha\_2” dla kontrahentów (odbiorców, dostawców, magazynów)

X - (od wersji 3.5) dana z pola „cecha\_3” dla kontrahentów (odbiorców, dostawców, magazynów)

W - (od wersji 3.5) dana z pola „cecha\_4” dla kontrahentów (odbiorców, dostawców, magazynów)

R - (od wersji 3.5) dana z pola „rabat\_marza\_narz\_procent” dla kontrahentów (odbiorców, dostawców, magazynów)

Na podstawie wpisanych parametrów tworzona jest przez program tablica tymczasowa TMP\_WYB, która zawiera klucze wybranych podczas „ptaszkowania” elementów. Po wykorzystaniu tablica TMP\_WYB jest usuwana przez program i nie należy się o to troszczyć.

Struktura tablicy TMP\_WYB:

typ	CHAR
numer	INT
indeks_znakowy	VARCHAR(80)

Sposób wypełniania przez program tej tablicy w zależności od typu (wpisywanego do pola „typ” tej tablicy):

*D* – numer=numer dostawcy, indeks\_znakowy jest nieistotny

*Z* – numer=numer dostawcy, indeks\_znakowy jest nieistotny

*k* – numer=numer rodzaju dokumentu płatności lub 1 dla paragonu (dane opisywane w tabeli dane\_dok\_plat), indeks\_znakowy jest nieistotny

*K* - numer=numer rodzaju dokumentu płatności (dane opisywane w tabeli dane\_dok\_plat), indeks\_znakowy jest nieistotny

*P* - numer=numer rodzaju dokumentu towarowego (dane opisywane w tabeli dane\_dok\_plat), indeks\_znakowy jest nieistotny

*U* - numer=numer użytkownika, indeks\_znakowy jest nieistotny

*O* - numer=numer odbiorcy, indeks\_znakowy jest nieistotny

*m* - numer=numer magazynu, indeks\_znakowy jest nieistotny

*M* - numer=numer magazynu, indeks\_znakowy jest nieistotny

*F* - numer=numer magazynu, indeks\_znakowy jest nieistotny

*S* – numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej opisanej w pliku konfiguracyjnym lista\_i2.txt w drugiej linii, najczęściej „gr.gł.” lub „SWW”

*V* - numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla VAT

*G* - numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej opisanej w pliku konfiguracyjnym lista\_in.txt w czwartej linii, najczęściej „grupa towarowa”

*9* - numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej opisanej w pliku konfiguracyjnym lista\_i2.txt w pierwszej linii, najczęściej „miejsce w magazynie”

*7* - numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej opisanej w pliku konfiguracyjnym lista\_i2.txt w czwartej linii

*8* - numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej opisanej w pliku konfiguracyjnym lista\_i2.txt w piątej linii

*1,2,4,5* - numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej opisanej w pliku konfiguracyjnym lista\_in.txt w liniach numer (odpowiednio 1,2,4,5)

*Y* – numer nieistotny, indeks\_znakowy=zawartość pola „indeks” przy oglądaniu słowników (menu Inne/Inne katalogi) z poziomu MagAD dla danej „forma płatności”

Zawartość przykładowa tabeli TMP\_WYB:

typ	numer	indeks_znakowy
P	1	
P	2	
P	4	
Y		gotówką
Y		przelewem
O	1	



O	12	
M	1	
M	2	
M	4	

Dla parametrów z takiej tabeli przez zapytania mogą być wybierane paragony i faktury sprzedaży bez korekt do faktur, z magazynów 1 i 2, dla odbiorców nr 1 i 12.

Wybory dla poszczególnych typów parametrów tego rodzaju program zapamiętuje dla danego użytkownika programu, podobnie jak i w innych zestawieniach.

### **Trzeci typ parametrów – zmienne zarezerwowane.**

Nazw zmiennych zarezerwowanych nie wolno używać, jako wpisywanych do tabeli „dane\_menu\_zes” w polu „nazwa”.

Zmienne zarezerwowane:

1)

**numer\_magazynu\_petla** – taką nazwę umieszczamy wtedy, kiedy ma być ona w pętli, w której zbieramy dane z magazynów, określonych w TMP\_WYB z wpisem w polu typ='M' wypełniona numerem magazynu (na numer\_magazynu\_petla podstawiana jest wartość numeru magazynu)

pętla zaczyna się, gdy komenda SQL = MAGAZYNY\_POCZATEK\_PETLI

pętla kończy się, gdy komenda SQL = MAGAZYNY\_KONIEC\_PETLI

w pętli wykonują się komendy SQL zawarte pomiędzy tymi komendami

2) (od wersji 3.5)

**UZYTKAKTUALNY** - jest wymieniane na numer uruchamiającego zestawienie użytkownika; można tej stałej użyć do różnych celów, m.in. do sięgania do jego pliku roboczego (tablica

„dane\_uzytkown\_UZYTKAKTUALNY\_plikrob\_”, np. dla użytkownika nr 1: „dane\_uzytkown\_1\_plikrob\_”)

3) (od wersji 3.5)

**MAGAKTUALNY** - jest wymieniane na numer magazynu, który jest wybrany, jako aktualny; można tej stałej użyć do różnych celów, m.in. do sięgania do tablicy z danymi stanów magazynu (tablica

„dane\_MAGAKTUALNY\_stany\_mg”, np. dla magazynu nr 10: „dane\_10\_stany\_mg”)

4) (od wersji 3.5)

Wewnątrz pętli magazynowej możliwe jest wykonanie pętli z podwyższaniem zmiennej iteracji o 1.

Początek takiej pętli oznaczony jest wpisem komendy do wykonania:

### **FOR\_START**

Następnie wyszukiwana jest komenda **FOR\_END**, która oznacza koniec pętli.

W polu "warunek" dla komandy FOR\_END jest zapytanie, którego wynikiem musi być liczba - będzie to liczba iteracji w pętli FOR.

W pętli wykonywane są zapytania, w których tekst "zmienna\_iteracji\_for" zamieniany będzie na aktualną wartość zmiennej iteracji pętli.

Pętlę taką można wykorzystywać np. do pobierania z tablicy o nazwach pól P1, P2, P3...Pn kolejnych wartości z kolejnych pól. Pole w takim przypadku symbolizowane będzie w kodzie „Pzmienna\_iteracji\_for”.

### **Definiowanie zapytań, które dadzą w wyniku raport.**

Odpowiedniej pozycji menu, podobnie, jak parametry, przyporządkować można praktycznie nieograniczoną ilość zapytań, prowadzących do stworzenia raportu. Wykonane one zostaną w kolejności wg pola „linia”.

Jeśli w polu „linia” jest 0, komenda a takim wpisem nie zostanie wykonana.

Treść zapytań wpisujemy w polu „tresc”. Pola „nazwa” i „opis” pozostawiamy puste.

Od wersji 3.5 – pole opis powiązane jest z polem „warunek” - patrz dalej.

Jeżeli program, wykonując zapytania, dotrze do zapytania, w którym podane są nazwy parametrów, to zastąpi nazwy tych parametrów stałymi, które wpisaliśmy „a konto” tych parametrów.

Jeżeli program napotka komendę SQL = MAGAZYNY\_POCZATEK\_PETLI, to wszystkie komendy, zawarte pomiędzy tą komendą, a komendą SQL = MAGAZYNY\_KONIEC\_PETLI zostaną wykonane dla po kolei każdego magazynu, który został wybrany do zestawienia przy wyborze magazynów, jako typu parametru „M”. W takiej pętli korzystamy ze zmiennej zarezerwowanej „numer\_magazynu\_petla”.

Takich pętli może być dowolna ilość w naszej strukturze zapytań tworzących raport, ale nie mogą być one jedna wewnątrz drugiej.

Należy zwrócić uwagę na to, że jeśli chcemy np. wykonać pętlę dla innego zestawu magazynów – np. generowanego automatycznie, można wstawić do taglicy TMP\_WYB odpowiednie wartości dla typu „M”. Jeśli znajdują się tam jakieś, można te wcześniejsze zachować w innej tablicy, a po wykonaniu odpowiednich operacji przywrócić. To samo dotyczy innych parametrów z tablicy TMP\_WYB.

Od wersji 3.5:

Pole „warunek”, jeżeli zawiera wpis, musi być to zapytanie typu SELECT. Jeśli takie zapytanie zwraca jeden lub więcej rekordów i pierwsze pole pierwszego rekordu tego zapytania zwraca wartość, która wpisana jest do pola „opis”, to wtedy i tylko wtedy wykonywane jest zapytanie z pola „tresc”. Na przykład można w polu „warunek” umieścić zapytanie to tablic systemowych, czy istnieje tabela, której dotyczy zapytanie w kolumnie „tresc”, aby nie dopuścić do błędu wykonania zapytania z pola „tresc”.

**W środku pętli FOR lub/i „po magazynach” (czyli między komendami MAGAZYNY\_POCZATEK\_PETLI i MAGAZYNY\_KONIEC\_PETLI) nie wolno umieszczać zapytań warunkowych. Czyli np. należy przed wykonaniem takiej pętli usunąć z tablicy TMP\_WYB magazyny, których tabel nie ma w bazie z jakiegś przyczyny.**

Np. kasowanie tabeli, pod warunkiem, że tabela istnieje (typ\_par jest puste):

opis	typ_par	tresc	warunek
1		DROP TABLE INF_DO_ZAPAMIETANIA	SELECT 1 FROM PG_TABLES WHERE SCHEMANAME='public' AND TABLENAME='inf_do_zapamietania'

### **Sortowanie wyniku.**

Kiedy już tablica TMP\_Z\_WL wypełniona jest danymi, zapytanie SELECT \* FROM TMP\_Z\_WL, którego wynik wyświetlany jest jako raport na ekranie, sortowane jest wg pól tablicy TMP\_Z\_WL wpisanych w pole „tresc” tablicy „dane\_menu\_zes”. Program odczytuje to sortowanie dzięki temu, że sortowanie w polu „typ” tablicy „dane\_menu\_zes” ma wpis równy 3.

Jeżeli sortowanie jest jedno, w polu „linia” musi być wpisana cyfra 1.

Sortowań może być zdefiniowanych więcej, niż jedno, lecz nie więcej, niż 19. Program na taką sytuację zareaguje pytaniem, które z tych sortowań ma użyć w raporcie. Wtedy w polu „linia” muszą być wpisane ich kolejne numery. Numery te muszą być liczbami naturalnymi i muszą zaczynać się od 1 i narastać bez pomijania żadnej liczby, czyli np. 1,2,3... W przykładach wypełnienia tabeli DANE\_MENU\_ZES poniżej w drugim, bardziej złożonym raporcie zdefiniowane są dwa sortowania.

### **Komentarze do dalej umieszczonej tabeli z przykładowymi zapytaniami.**

W tabeli mamy zdefiniowane menu, w którym dla pozycji 1 i 3 z pierwszego poziomu menu zdefiniowane są parametry i zapytania tworzące raport (patrz – tabele na następnych dwóch stronach).

Pierwszy raport jest prosty – jest połączoną listą odbiorców i magazynów, gdzie odrzuca się odbiorców i magazyny o numerze określonym w parametrze „kontr\_numer”. Wynik sortowany jest wg nazw odbiorców i magazynów.

Drugi raport tworzy listę dokumentów o wybranych rodzajach, dla wybranych magazynów, a następnie ją podsumowuje. Całkowicie zbędnie pyta o numer kontrahenta od.. i do.., ponieważ potem nie korzysta z tych parametrów. W tym przykładzie widać dobrze, jak używać pętli, która pozwala wyluskiwać i łączyć dane z poszczególnych magazynów. Dodatkowo przedstawione jest, jak z zakodowanego w bazie MagAD numeru dokonać „wyciągnięcia” roku i numeru dokumentu. Wynik sortowany jest wg typów dokumentów, lat i numerów dokumentów.



Przykładowa zawartość tabeli dane\_menu\_zes:

uzytk	typ	menu0	menu1	menu2	linia	nazwa	opis	typ_par	tresc
0	0	1	0	0	1	Zestawienie nr 1	daje jeden		
0	0	2	0	0	2	Zestawienie nr 2	daje dwa		
0	0	2	1	0	11	Zestawienie nr 2 -	daje dwa - jeden		
0	0	2	2	0	22	Zestawienie nr 2 -	daje dwa - dwa		
0	0	2	2	1	12 -1	Zestawienie nr 2 -	daje dwa - dwa - jeden		
0	0	2	2	2	22 -2	Zestawienie nr 2 -	daje dwa - dwa - dwa		
0	0	2	2	3	32 -3	Zestawienie nr 2 -	daje dwa - dwa - trzy		
0	0	2	3	0	33	Zestawienie nr 2 -	daje dwa - trzy		
0	0	2	4	0	44	Zestawienie nr 2 -	daje dwa - cztery		
0	0	3	0	0	3	Zestawienie nr 3	daje trzy		
0	1	1	0	0	1	kontr_numer	Podaj numer kontrahenta, którego należy usunąć z zestawienia:	03	
0	1	3	0	0	0	odb0	Wybierz odbiorców:	0	
0	1	3	0	0	1	nr0	Podaj numer kontrahenta od:	03	
0	1	3	0	0	2	nr1	Podaj numer kontrahenta do:	03	
0	1	3	0	0	3	dok0	Wybierz typy dokumentów:	P	
0	1	3	0	0	5	mag0	Wybierz magazyny:	M	
0	2	1	0	0	1				CREATE TEMPORARY TABLE TMP_Z_WL AS SELECT *FROM DANE_MAGAZYNY
0	2	1	0	0	2				INSERT INTO TMP_Z_WL SELECT *FROM DANE_ODBIORCY
0	2	1	0	0	3				DELETE FROM TMP_Z_WL WHERE NUMER=kontr_numer
0	2	3	0	0	1				CREATE TEMPORARY TABLE TMP_Z_WL(TYP_DOK SMALLINT,ROK_DOK INT,NR_DOK INT,ZNAK_DOK VARCHAR(4),SUMA_DOK NUMERIC(28,8))
0	2	3	0	0	2				MAGAZYNY_POCZATEK_PETLI
0	2	3	0	0	3				INSERT INTO TMP_Z_WL SELECT TYP_DOK,1996+(NR_DOK>>24),NR_DOK- ((NR_DOK>>24)<<24),ZNAK_DOK,-SUMA_DOK FROM DANE_numer_magazynu_petla_NAGLOWKI WHERE TYP_DOK IN (SELECT NUMER FROM TMP_WYB WHERE TYP=P') AND TYP_DOK IN (1,2,4,8,9,13,14,15)
0	2	3	0	0	4				INSERT INTO TMP_Z_WL SELECT TYP_DOK,1996+(NR_DOK>>24),NR_DOK- ((NR_DOK>>24)<<24),ZNAK_DOK,SUMA_DOK FROM DANE_numer_magazynu_petla_NAGLOWKI WHERE TYP_DOK IN (SELECT NUMER FROM TMP_WYB WHERE TYP=P') AND TYP_DOK IN (3,5,6,7,10,11,12,16,17,18)
0	2	3	0	0	5				MAGAZYNY_KONIEC_PETLI
0	2	3	0	0	6				INSERT INTO TMP_Z_WL SELECT 30000,0,0,'sum:',SUM(SUMA_DOK) FROM TMP_Z_WL
0	3	1	0	0	1				NAZWA
0	3	3	0	0	1				TYP_DOK,ROK_DOK,NR_DOK,ZNAK_DOK
0	3	3	0	0	2				SUMA_DOK

Zawartość tabeli dane\_menu\_zes z wyselekcjonowanymi tylko elementami pierwszego raportu (czyli uzytk=0 i menu0=1 i menu1=0 i menu2=0):

uzytk	typ	menu0	menu1	menu2	linia	nazwa	opis	typ_par	tresc
0	0	1	0	0	1	Zestawienie nr 1	daje jeden		
0	1	1	0	0	1	kontr_numer	Podaj numer kontrahenta, którego należy usunąć z zestawienia:	03	
0	2	1	0	0	1				CREATE TEMPORARY TABLE TMP_Z_WL AS SELECT * FROM DANE_MAGAZYN
0	2	1	0	0	2				INSERT INTO TMP_Z_WL SELECT * FROM DANE_ODBIORCY
0	2	1	0	0	3				DELETE FROM TMP_Z_WL WHERE NUMER=kontr_numer
0	3	1	0	0	1				NAZWA

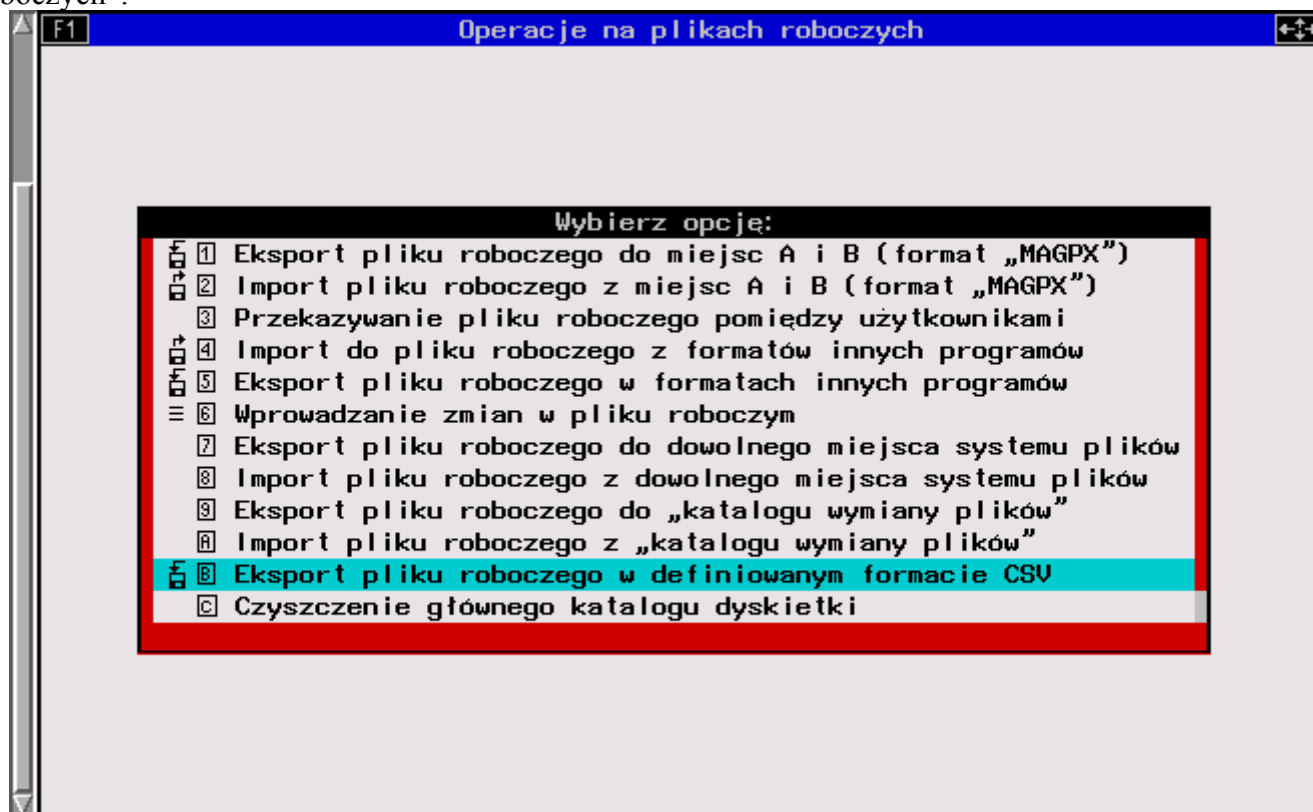
Zawartość tabeli dane\_menu\_zes z wyselekcjonowanymi tylko elementami drugiego raportu (czyli uzytk=0 i menu0=3 i menu1=0 i menu2=0):

uzytk	typ	menu0	menu1	menu2	linia	nazwa	opis	typ_par	tresc
0	0	3	0	0	3	Zestawienie nr 3	daje trzy		
0	1	3	0	0	0	odb0	Wybierz odbiorców:	0	
0	1	3	0	0	1	nr0	Podaj numer kontrahenta od:	03	
0	1	3	0	0	2	nr1	Podaj numer kontrahenta do:	03	
0	1	3	0	0	3	dok0	Wybierz typy dokumentów:	P	
0	1	3	0	0	5	mag0	Wybierz magazyny:	M	
0	2	3	0	0	1				CREATE TEMPORARY TABLE TMP_Z_WL(TYP_DOK SMALLINT,ROK_DOK INT,NR_DOK INT,ZNAK_DOK VARCHAR(4),SUMA_DOK NUMERIC(28,8))
0	2	3	0	0	2				MAGAZYNY_POCZATEK_PETLI
0	2	3	0	0	3				INSERT INTO TMP_Z_WL SELECT TYP_DOK,1996+(NR_DOK>>24),NR_DOK- ((NR_DOK>>24)<<24),ZNAK_DOK,-SUMA_DOK FROM DANE_numer_magazynu_petla_NAGLOWKI WHERE TYP_DOK IN (SELECT NUMER FROM TMP_WYB WHERE TYP=P') AND TYP_DOK IN (1,2,4,8,9,13,14,15)
0	2	3	0	0	4				INSERT INTO TMP_Z_WL SELECT TYP_DOK,1996+(NR_DOK>>24),NR_DOK- ((NR_DOK>>24)<<24),ZNAK_DOK,SUMA_DOK FROM DANE_numer_magazynu_petla_NAGLOWKI WHERE TYP_DOK IN (SELECT NUMER FROM TMP_WYB WHERE TYP=P') AND TYP_DOK IN (3,5,6,7,10,11,12,16,17,18)
0	2	3	0	0	5				MAGAZYNY_KONIEC_PETLI
0	2	3	0	0	6				INSERT INTO TMP_Z_WL SELECT 30000,0,0,'sum:',SUM(SUMA_DOK) FROM TMP_Z_WL
0	3	3	0	0	1				TYP_DOK,ROK_DOK,NR_DOK,ZNAK_DOK
0	3	3	0	0	2				SUMA_DOK

## Eksporty danych z pliku roboczego, które można zdefiniować.

Program potrafi wyeksportować dane z pliku roboczego do pliku CSV według odpowiedniej definicji. Dla niewtajemniczonych – CSV to dane eksportowane w pliku tekstowym, wartości w kolumnach oddzielane są specjalnym znakiem. W tym przypadku znakiem tym jest średnik. Warto nadmienić, że dzielnikiem dziesiętnym w CSV generowanym z MagAD jest kropka, czyli w danych CSV będą liczby typu: „456.56”, a nie „456,56”. Eksport taki zawiera zawsze w pierwszej linii nazwy kolumn.

Opcja eksportu pliku roboczego do CSV dostępna jest z poziomu funkcji „Towary/Operacje na plikach roboczych”:



Od wersji 3.4.8-9 definicji może być do 800 sztuk (rodzajów). Zatem o ile jakiś eksport jest zdefiniowany, pokaże się menu, zawierające listę zdefiniowanych, dostępnych eksportów:

**Wybierz rodzaj eksportu do pliku CSV:**  
Eksport wybranego dokumentu dla Kadeta  
eksport na WWW  
Esc-wyjście/rezygnacja

W niższych wersjach MagAD-SQL wolno definiować tylko jeden eksport.

Następnie program poprosi o podanie nazwy pliku wynikowego. Kiedy program proponuje nazwę pliku, naciśnięcie klawisza F1 wywoła listę wyboru. Zawiera ona propozycje, powodujące wysłanie pliku do miejsc A, B i katalogu wymiany MagAD. Np.: "b/wynik.csv" to miejsce B. W systemach Linux przy standardowej instalacji MagAD miejscem B jest pulpit.

Program proponuje kodowania polskich znaków w takim pliku:

- 1 ISO8859-2
- 2 WinCP1250
- 3 DOScp852
- 4 Mazovia
- 5 UTF8 (od wersji MagAD 3.5.x)
- 6 zamienniki angielskie, czyli bez polskich znaków, zamiast Å jest A itp. (od wersji MagAD 3.5.x)

Należy spytać się odbiorcy, który standard mu odpowiada. Warto go też poinformować, że znakiem dzielącym pole jest w średnik, a teksty ujmowane są w znaki podwójnego górnego cudzysłowu.

Excel akceptuje plik z kodowaniem nr 2, OpenOffice/LibreOffice kodowania od 1 do 3.

Definicję może wykonać informatyk, który zna język operacji na bazach danych (SQL). Od wersji 3.4.8-9 definicji może być do 800 sztuk (rodzajów).

Opis techniczny:

*Definicje eksportów znajdują się w tabeli:*

Table "public.dane\_mplr\_csv":

Column	Type	Modifiers	Opis
nr_def	integer	not null	numer kolejny definicji
nazwa	character varying(25)	not null	nazwa definicji
id_w_def	double precision	not null	lp zapytania w definicji
tresc	text		kod SQL zapytania
kod_pl	smallint	default 0	numer kodowania polskich znaków

Indexes:

"dane\_mplr\_csv\_pkey" PRIMARY KEY, btree (nr\_def, nazwa, id\_w\_def)

"dane\_mplr\_csv\_oidindeks" UNIQUE, btree (oid)

Numery kodowania polskich znaków:

1 ISO8859-2

2 WinCP1250

3 DOScp852

4 Mazovia

5 UTF-8

6 zamienniki angielskie.

Widoczne powyżej opcje 5 i 6 dostępne są od wersji 3.5.x. Jeżeli w polu „kod\_pl” wpisane jest 0, program zapyta o kodowanie.

*Tabela ta musi być założona za pomocą programu do upgrade baz, aby we właściwy sposób została zarejestrowana do obsługi w MagAD. Zawartość tabeli można edytować za pomocą dowolnego narzędzia, w którym jest możliwa edycja tabel PostgreSQL (np.psql, PgAdmin, OpenOffice, MS Access).*

*Definiowanie eksportu pliku roboczego jest proste w porównaniu do definiowania raportów.*

*W jednej definicji może być wiele zapytań, które utworzą tabelę tymczasową TMP\_Z\_WL. eksportowany jest wynik zapytania: select \* from TMP\_Z\_WL. Po eksporcie tabela jest usuwana przez program.*

*Tabela TMP\_Z\_WL musi być zakładana, jako tymczasowa. W innym wypadku wystąpią konflikty, gdy wielu użytkowników będzie chciało wykonać eksporty. Jeżeli błąd stworzenia TMP\_Z\_WL popełnimy także gdzieś w definiowaniu raportów, wystąpią także tego typu problemy.*

*Plik roboczy użytkownika w zapytaniach ma symboliczną nazwę PLIKROB\_. Kiedy zapytania są realizowane przez program, nazwa ta zastępowana jest nazwą tablicy, która jest „plikiem roboczym użytkownika”. Zapytania dla danej definicji realizowane są w kolejności wg pola „id\_w\_def”.*

*Oto przykładowa zawartość tablicy „dane\_mplr\_csv”:*

nr_def	nazwa	id_w_def	tresc
1	całość pliku roboczego	1	create temporary table tmp_z_wl as select * from PLIKROB_
2	sklejenie pl.rob.Kierownika Systemu z pl.rob.aktualnego użytkownika	1	create temporary table tmp_z_wl as select * from PLIKROB_
2	sklejenie pl.rob.Kierownika Systemu z pl.rob.aktualnego użytkownika	2	insert into tmp_z_wl select * from dane_uzytkown_0_plikrob_

Proszę zwrócić uwagę, na celowe użycie małych liter w nazwie tabeli „*dane\_uzytkown\_0\_plikrob\_*”. Gdyby napisać to wielkimi literami, słowo „*PLIKROB\_*” zastąpione by zostało nazwą tabeli pliku roboczego aktualnie pracującego użytkownika.

Od wersji 3.5.3:

Jeżeli eksport wywoływany jest z opcji edycji zawartości towarowej (specyfikacji) dokumentu, to zmienna *PLIKROB\_* oznacza nie plik roboczy, ale aktualnie obrabianą specyfikację dokumentu.

*PLIKROBABS\_* oznacza zawsze plik roboczy, niezależnie od opcji, z której się wywołało eksport

*PRACAABS\_* oznacza zawsze aktualnie obrabiana specyfikację dokumentu, niezależnie od opcji, z której się wywołało eksport

Dodatkowo program na zmiennych, podanych poniżej przenosi:

*NRMAGAKT\_* - numer aktualnego magazynu

*KATMAGAKT\_* - symbol magazynu w nazwach tablic, takich, jak „*dane\_KATMAGAKT\_naglowki*” - dla numerów do 8 cyfr jest taki sam, jak *NRMAGAKT\_*

*TYPDOK\_* - numer typu dokumentu (patrz zawartość tablicy „*dane\_dok\_plat*”).

*NRDOK\_* - numer dokumentu (taki, jak w bazie danych, w jednej licznie posiadający zakodowany rok dokumentu i numer w roku)

*ZNDOK\_* - rozszerzenie tekstowe numeru „znak dokumentu” (do 4 znaków pisarskich)

*DATADOK\_* - data dokumentu

*NRKON\_* - numer kontrahenta dokumentu (w zależności od typu dokumentu jest to odbiorca, dostawca lub magazyn)

*FORMPLAT\_* - skrót formy płatności (dana pobrana z pola „*sposob\_rozlicz*” z tablicy nagłówków dokumentów - *dane\_KATMAGAKT\_naglowki*).

Parametry *TYPDOK\_*, *NRDOK\_*, *ZNDOK\_*, *DATADOK\_*, *NRKON\_*, *FORMPLAT\_* sprawdzane są zawsze przy wywołaniu z poziomu „Wprowadzenia i zmian”/edycja zawartości dokumentu. Z poziomu operacji na plikach roboczych tylko wtedy, gdy operator odpowie TAK na pytanie, czy powiązać plik roboczy z dokumentem.