

WAPRO ERP

by **ASSECO**

Procedury SQL obsługi dokumentów w WAPRO Mag. Poradnik wdrożeniowca.

obowiązuje od wersji 8.50.0

Opracował:
Rafał Mróz

Wykonał:
Grzegorz Lenarczyk

Asseco Business Solutions SA
Oddział w Warszawie
ul. Adama Branickiego 13
02-972 Warszawa
www.assecobs.pl/wapro

Copyright © Asseco Business Solutions SA

Spis treści

Wstęp	4
1. Procedury dotyczące dokumentu magazynowego	4
1.1 Procedura zatwierdzająca nagłówek dokumentu	4
1.2 Dodanie pozycji dokumentu	5
1.2.1 Pomocnicza procedura obliczająca ilość w jednostce magazynowania dla wskazanej „innej jednostki”	8
1.3 Procedura zatwierdzająca dokument	8
1.3.1 Procedura JL_SumujDokumentMagazynowy_Server	9
1.3.2 Procedura JL_PobierzFormatNumeracji_Server	9
1.3.3 Procedura zatwierdzająca dokument magazynowy	10
1.4 Procedura usuwająca dokument magazynowy	13
1.5 Przykładowa procedura zatwierdzająca dokument magazynowy	14
2. Procedury dodające dokument handlowy	17
2.1 Dodanie nagłówka dokumentu handlowego	18
2.1.1 Dodanie nagłówka dokumentu handlowego – inicjacja wprowadzania dokumentu	18
2.1.2 Procedura JL_PobierzFormatNumeracji_Server	19
2.1.3 Automatyczne utworzenie dokumentu magazynowego	19
2.1.4 Pomocnicza procedura obliczająca ilość w jednostce magazynowania dla wskazanej „innej jednostki”	21
2.2 Procedura zatwierdzająca pozycję dokumentu handlowego jednocześnie do dokumentu magazynowego i handlowego	22
2.3 Procedura zatwierdzająca dokument handlowy	24
2.3.1 Sumowanie dokumentu handlowego procedurą JL_SumujDokumentHandlowy_Server	24
2.3.2 Procedura JL_PobierzFormatNumeracji_Server	25
2.3.3 Procedura zatwierdzania dokumentu handlowego	26
2.3.4 Procedura generująca dokument finansowy do zatwierdzonej faktury AP_GenerujRozliczenieDokumentuHandlowego	30
2.4 Wyjątki – dla procedury zatwierdzającej dokument handlowy – korekta	31
2.4.1 Procedura dodająca dokument handlowy korekty	31
2.4.2 Korygowanie pozycji	32
2.4.3 Procedura do zatwierdzania dokumentu korekty	34
2.4.4 Procedura rozliczająca dokument korekty	34
2.5 Przykład dodania dokumentu handlowego FZUE	34
3. Dodanie dokumentu finansowego	44
3.1 Procedura dodająca nagłówek dokumentu finansowego	44
3.2 Procedura AP_DodajPozycjeDlaDokFin	45
3.3 Procedura AP_WciagnijRozrachunkiNaPozycje	46

3.4	Zatwierdzanie dokumentu finansowego.....	46
3.4.1	Procedura JL_PobierzFormatNumeracji_Server.....	46
3.4.2	Procedura sprawdzająca rozrachunki do rozliczenia	47
3.4.3	Procedura zatwierdzająca dokument finansowy.....	48
3.5	Przykłady dodawania dokumentów finansowych.....	49
4.	Dodawanie zamówień.....	51
4.1	Procedura dodająca nagłówek zamówienia.....	51
4.2	Procedura dodawania pozycji zamówienia.....	52
4.3	Zatwierdzanie zamówienia.....	54
4.3.1	Procedura sumująca zamówienie	54
4.3.2	Procedura JL_PobierzFormatNumeracji_Server.....	55
4.3.3	Procedura zatwierdzająca zamówienie.....	56
5.	Zamykanie oraz anulowanie zamówienia.....	58
5.1	Anulowanie /zamykanie zamówień – opis procedury	58
5.2	Wycofanie Anulowania/zamknięcia – opis procedury	59
5.3	Przykład dodawania zamówienia	59
6.	Dodawanie asortymentu	63
7.	Dodawanie kontrahenta	63
8.	Procedury pomocnicze dokumentów.....	63
8.1	Funkcja obliczająca wartość brutto dbo.JL_Func_ObliczBrutto()	63
8.2	Funkcja obliczająca wartość netto dbo.JL_Func_ObliczNetto()	64
8.3	Pobieranie parametrów z konfiguracji firmy - dbo.AP_PobierzParaKonf()	64
8.4	Dodawanie informacji do logu aktywności dbo.AP_ZapiszDoLogu ()	65
8.5	Sumowanie korekty	66
9.	Wykorzystanie tabeli zaznaczone.	67
10.	Dodawanie dokumentu zlecenia produkcyjnego	67
10.1	Dodawanie nagłówka zlecenia produkcyjnego	68
10.2	Dodawanie pozycji zlecenia produkcyjnego.....	68
10.3	Procedura JL_PobierzFormatNumeracji_Server.....	69
10.4	Zatwierdzanie zlecenia produkcyjnego	70
10.5	Przykładowe wywołanie procedur do założenia zlecenia produkcyjnego montażu.....	72
11.	Dodawanie dokumentu oferty.	75
11.1	Procedura dodawania nagłówka oferty	75
11.2	Procedura dodawania pozycji oferty	75
11.3	Procedura zatwierdzająca dokument oferty	77
11.4	Przykładowe wywołanie procedur zakładających dokument oferty.....	78
12.	Zmiany w dokumencie.....	81
12.1	Poprawki w dokumentacji (2012-10-25).....	81
12.2	Poprawki w dokumentacji (2015-07-01).....	81
12.3	Poprawki w dokumentacji (2016-01-20).....	81

12.4 Poprawki w dokumentacji (2017-03-10).....	81
a) Dostosowanie do wersji 8.21	81
12.5 Poprawki w dokumentacji (2018-10-10).....	81
a) Dostosowanie do wersji 8.40	81
12.6 Poprawki w dokumentacji (2018-10-25).....	82

Wstęp.

Dokumentacja procedur pozwalająca wykorzystać dodawanie, usuwanie dokumentów zamówień, magazynowych, handlowych, finansowych, zleceń produkcyjnych, kartotek przy pomocy mechanizmów API WAPRO Mag.

1. Procedury dotyczące dokumentu magazynowego.

Większość danych i obsługę dokumentów realizują procedury składowane. Dla pól, które nie występują jako parametry tych procedur należy wykonać standardowe zdanie SQL:

```
Update Dokument_magazynowy
  Set <nazwa pola> = <wartość>
Where id_dok_magazynowego = <unikalny w bazie danych
identyfikator dokumentu>
```

Powyższy Update będzie stosowany najczęściej dla pól dodatkowych obsługiwanych w WAPRO Mag Prestiż i Prestiż Plus.

1.1 Procedura zatwierdzająca nagłówki dokumentu

Do zatwierdzania dokumentu magazynowego można użyć procedury AP_DodajWTleDokMag_Server. Tej samej procedury używamy podczas dodawania dokumentu handlowego.

Procedura i jej parametry.

```
Exec AP_DodajWTleDokMag_Server
@id_firmy,@id_dok_handl,@id_mag,@Data,@id_typu,@id_kontrahenta,@przychod,@rozchod,
@rodzaj,@wycena,@brutto_netto,@id_uzytkownika, @Id_dokumentu output
```

Opis parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Numeric		Id_Firmy z tabeli firma
2.	@Id_dok_handl	Numeric		0 – w przypadku dodawania dokumentu magazynowego
3.	@id_magazynu	Numeric		id_magazynu z tabeli magazyn
4.	@Data	Int		round(convert(real,getdate()),0) + 36163
5.	@id_typu	Int		id_typu z tabeli - typ_dokumentu_magazynowego
6.	@id_kontr	Numeric		id_kontrahenta z tabeli kontrahent
7.	@przychod	Tinyint		Jeśli dokument przychodowy wartość 1, dla rozchodowego 0
8.	@rozchod	Tinyint		Jeśli dokument przychodowy wartość 0 dla rozchodowego 1
9.	@rodzaj	Char(2)		'WZ' dla rozchodowych i 'PZ' dla przychodowych, 'MW' dla dok. MW.
10.	@wycena	Char(1)		'Z' wg cen zakupu lub 'S' wg cen sprzedaży
11.	@brutto_netto	Char(6)		Obliczanie dokumentu wg - 'Netto' lub 'Brutto'
12.	@id_uzy	Numeric(18,0)		id użytkownika zakładającego dokument z tabeli użytkownik.
13.	@id_dokumentu	Numeric(18,0)		Parametr OUTPUT, zwraca id_dokumentu – id nagłówka w bazie danych – id_dok_magazynowego z tabeli dokument magazynowy

1.2 Dodanie pozycji dokumentu.

Procedura i jej Parametry

Exec JL_ZatwierdzPozycje_Server
 @id_dok_mag,@id_dok_handl,@id_artykulu,@rodzaj_pozycji,@rodzaj_artykulu,@jednostka,
 @przelicznik,@ilosc,@kod_vat,@cena_netto,@cena_brutto,
 @cena_netto_wal,@cena_brutto_wal,
 @rabat,@znak_rabatu,@rabat2,@znak_rabatu2,
 @fifo_lifo,@ostrzegaj_stan_zero,@data,@wycena,
 @opakowania_wydano,@opakowania_przyjeto,@pozycja_drs

@nr_paczki, ,@Oblicz_wartosc_pozycji, @tryb_rejestracji,@id_poz_dostawy,
 @wartosc_pozycji_netto, @wartosc_pozycji_brutto,@r_id_poz_dok, @tryb_cichy,
 @err_cichy, @zn_ceny, @czb_marza, @nr_serii_we, @data_waznosci_we

Opis Parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@ id_dok_mag	Numeric		Wewnętrzne pole dokumentu (pole ma właściwość IDENTITY) zwracane przez JL_DodajDokumentMagazynowy
2.	@id_dok_handl	Numeric	0	0 – jeśli pozycja nie powiązana z dokumentem handlowym
3.	@id_art	Numeric		id_artykułu na pozycję z tabeli artykuł
4.	@rodzaj_poz	Char(1)		'R' dla rozchodu lub 'P' dla przychodu
5.	@rodzaj_art	Char(1)		Pierwszy znak z pola rodzaj z tabeli artykuł np. T- Towar, P-Produkt, U-Usługa, itd.
6.	@jednostka	Varchar(10)		skrót jednostki z tabeli jednostka
7.	@przelicznik	Decimal(16,6)		przelicznik jednostki z tabeli jednostka
8.	@ilosc	Decimal(16,6)		ilość asortymentu na pozycji w jednostce magazynowania
9.	@kod_vat	Char(3)		kod stawki VAT z tabeli stawka_vat
10.	@cena_netto	Decimal(14,2)		Cena netto
11.	@cena_brutto	Decimal(14,2)		Cena brutto
12.	@cena_netto_wal	Decimal(14,4)	0 – dla dokumentów magazynowych	Cena netto w walucie
13.	@cena_brutto_wal	Decimal(14,4)	0 – dla dokumentów magazynowych	Cena brutto w walucie
14.	@rabat	Decimal(8,2)		Procent rabatu lub narzutu
15.	@znak	Tinyint		znak rabat 1 kwoty dla parametru - @rabatu >=0 wartość 2 w przeciwnym wypadku 0
16.	@rabat2	Decimal(8,2)	0 – dla dokumentów magazynowych	Procent rabatu z dokumentu handlowego

17.	@znak2	Tinyint		znak rabat 2 kwoty dla parametru - @rabatu2 >=0 wartość 2 w przeciwnym wypadku 0
18.	@fifo_lifo	Char(4)		wartości 'FIFO' lub 'LIFO'
19.	@ostrzegaj_stan_zero	Tinyint		0 nie ostrzegaj lub 1 ostrzegaj
20.	@data	Int		round(convert(real,getdate()),0) + 36163
21.	@wycena	Char(1)		Z' wg cen zakupu lub 'S' wg cen sprzedaży
22.	@op_wydano	Decimal(14,4)	0 – jeśli towar nie jest opakowaniem	Pola używane dla asortymentu opakowanie
23.	@op_przyjeto	Decimal(14,4)	0 – jeśli towar nie jest opakowaniem	Pola używane dla asortymentu opakowanie
24.	@pozycja_drs	Tinyint	0	
25.	@nr_paczki	Varchar(20)	''	opis klasy towaru wymagany dla faktur RR – bez znaczenia w pozostałych przypadkach opis klasy towaru wymagany dla faktur RR – bez znaczenia w pozostałych przypadkach
26.	@Oblicz_wartosc_pozycji	Tinyint	0	Czy obliczyć wartość pozycji 0 – nie , 1 – tak, uzupełniane są wtedy zmienne w punkcie 29 oraz 30
27.	@tryb_rejestracji	Tinyint	0	Pole wewnętrzne.
28.	@id_poz_dostawy	Numeric	0	jeżeli rozchód jest robiony z dostawy to należy przekazać id_poz_dok_mag z wybranej pozycji przychodowej
29.	@wartosc_pozycji_netto	Decimal(14,2)		Parametr OUTPUT Wartość netto pozycji
30.	@wartosc_pozycji_brutto	Decimal(14,2)		Parametr OUTPUT Wartość brutto pozycji
31.	@r_id_poz_dok	Numeric(18)		Parametr OUTPUT – zwraca id_poz_dok_mag zapisanej pozycji
32.	@tryb_cichy	tinyint	0	Jeśli 0 to brak komunikatu jeśli 1 to zwrócona wartości do @err_cichy
33.	@err_cichy	Tinyint		Błąd zwrócony podczas zapisywania pozycji - parametr OUTPUT
34.	@zn_ceny	Char(1)	k	k kartotekowa p promocyjna i indywidualna g dla grupy cenowej m modyfikowana ręcznie
35.	@czb_marza	Decimal(14,4)	0	>0 gdy zatwierdzamy pozycje dokumentu rozchodu podlegającego VAT marża

36.	@nr_serii_we	Varchar(50)	''	numer serii preferowanej dostawy z której ma nastąpić rozchód. Parametr używany przy ustawieniu w kartotece asortymentowej metody rozchodu wg numeru serii/daty ważności
37.	@data_waznosci_we	Int	0	data ważności preferowanej dostawy z której ma nastąpić rozchód (analogicznie jak powyżej)

1.2.1 Pomocnicza procedura obliczająca ilość w jednostce magazynowania dla wskazanej „innej jednostki”

Procedura i jej parametry

EXEC AP_PobierzJednostkeArtykulu @id_jed, @id_fir, @id_art, @skrot

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_jed	Numeric		Wewnętrzne id jednostek
2.	@id_fir	Numeric		identyfikator firmy z tabeli FIRMA, w której rejestrowany jest dokument
3.	@id_art	Numeric		id artykułu z tabeli artykul
4.	@skrot	Varchar(10)		Skrót jednostki, dla której ma być obliczona ilość w jednostce magazynowania

Przykładowe wywołanie

EXEC AP_PobierzJednostkeArtykulu 2,1,34,'opk.'

1.3 Procedura zatwierdzająca dokument

Do poprawnego zatwierdzenia dokumentu potrzebne jest kolejne wywołanie procedur najpierw JL_PobierzFormatNumeracji_Server , JL_SumujDokumentMagazynowy_Server, i następnie JL_ZatwierdzDokumentMagazynowy_Server

1.3.1 Procedura JL_SumujDokumentMagazynowy_Server

Procedura i jej parametry

```
EXEC JL_SumujDokumentMagazynowy_Server  
@id_dok_mag ,  
@rodzaj_dok,  
@Brutto_netto,  
@nr_filtra,  
@Suma_netto OUTPUT,  
@Suma_brutto OUTPUT
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1	@id_dok_mag	Int		Id_dok_magazynowego z tabeli dokument_magazynowy
2	@rodzaj_dok	Char(2)		Dla dokumentów przychodowych 'PZ' dla rozchodowych 'WZ', 'MW' oraz 'MP' w przypadku sumowania dokumentów MM
3	@brutto_netto	Char(6)		Ewidencja dokumentu 'Netto' lub 'Brutto'
4	@nr_filtra	Int	1	

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury JL_ZatwierdzDokumentMagazynowy_Server jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze JL_ZatwierdzDokumentMagazynowy_Server
1	@suma_netto	Decimal(16,6)	@suma_netto
2	@suma_brutto	Decimal(16,6)	@suma_brutto

1.3.2 Procedura JL_PobierzFormatNumeracji_Server

Procedura i jej parametry

```
EXEC JL_PobierzFormatNumeracji_Server
```

@id_firmy ,
 @dokument,
 @id_typu,
 @id_zasobu ,
 @format_num OUTPUT,
 @okresnumeracji OUTPUT,
 @parametr1 OUTPUT,
 @parametr2 OUTPUT

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Int		Id_firmy z tabeli firma
2.	@dokument	Tinyint	1	
3.	@id_typu	Int		Id_typu z tabeli typ_dokumentu_magazynowego
4.	@id_zasobu	Int		id_magazynu z tabeli magazyn

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury JL_ZatwierdzDokumentMagazynowy jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze JL_zatwierdzDokumentMagazynowy
1.	@format_numeracji	Varchar(50)	@num_format
2.	@okresnumeracji	Tinyint	@num_okresnumeracji
3.	@Parametr1	Tinyint	@num_auto
4.	@Parametr2	TinyInt	@num_niezalezny

1.3.3 Procedura zatwierdzająca dokument magazynowy

Procedura i jej parametry

exec JL_ZatwierdzDokumentMagazynowy_Server

@id_dok_mag , @id_typu , @rodzaj_dok , @data ,
@id_kontr , @id_mag_doc , @przychod , @rozchod ,
@suma_netto , @suma_brutto , @id_dok_handl , @numer ,
@num_format , @num_okresnumeracji @num_auto , @num_niezalezny ,
@id_firmy , @id_mag , @brutto_netto , @dh_suma_netto ,
@dh_suma_brutto , @przeliczaj_ceny , @dodaj_ceny_MM , @aktualizuj_ceny_sprz_MM ,
@id_kasy , @wpłata_biezaca , @zaliczka_d , @id_uzytkownika ,
@id_pracownika , @uwagi , @ro_id , @ro_forma , @ro_termin , @Data_VAT

Opis Parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_dok_mag	Numeric		wewnętrzne id dokumentu
2.	@id_typu	Numeric		z tabeli typ_dokumentu_magazynowego
3.	@rodzaj_dok	Char(3)		'WZ' dla rozchodowych i 'PZ' dla przychodowych, 'MW' dla dokumentu przesunięcia MM
4.	@data	Int		round(convert(real,getdate()),0) + 36163
5.	@id_kontr	Numeric		Id_kontrahenta z tabeli kontrahent, w przypadku dokumentu MM 0
6.	@id_mag_doc	Numeric	0	id magazynu w przypadku dokumentów MM
7.	@przychod	Tinyint		wartość 0 dla rozchodowego lub 1 dla przychodowego
8.	@rozchod	Tinyint		wartość 1 dla rozchodowego lub 0 dla przychodowego
9.	@suma_netto	Decimal(14,2)		wartość netto dokumentu , pobrać sumę z <u>jl_sumujdokumentmagazynowy_server</u>
10	@suma_brutto	Decimal(14,2)		wartość brutto dokumentu, pobrać sumę z <u>jl_sumujdokumentmagazynowy_server</u>
11	@id_dok_handl	Numeric		0 dla dokumentów magazynowych bez dokumentów handlowych
12	@numer	Varchar(30)		numer dokumentu
13	@num_format	Varchar(30)		maska formatu numeracji,
14	@num_okresnumeracji	Tinyint		ciągły, miesięczny, roczny, dzienny , wartości (1,0,2,3)
15	@num_auto	Tinyint		numeracja automatyczna lub ręczna (1,2)
16	@num_niezalezny	Tinyint		numeracja zależna od magazynu lub niezależna (1- zależna , 2 -niezależna)

17	@id_firmy	Int		pole id_firmy z tabeli firma
18	@id_mag	Int		pole id_magazynu z tabeli magazyn
19	@brutto_netto	Char(6)		'Netto' lub 'Brutto'
20	@dh_suma_netto	Decimal(14,2)	0	
21	@dh_suma_brutto	Decimal(14,2)	0	
22	@przeliczaj_ceny	Tinyint	0	parametr ważny dla dokumentów przychodowych, 1 – zaktualizuje ceny zakupu (zmiana marży), 2- zaktualizuje ceny zakupu i sprzedaży w kartotece asortymentowej (marża bez zmian), 0 nie ruszaj cen
23	@dodaj_ceny_MM	Tinyint	0	parametr ważny dla dok MM, 1 = jeśli dodawaj ceny w magazynie docelowym
24	@aktualizuj_ceny_sprz_MM	Tinyint	0	parametr ważny dla dok MM – 1 jeśli aktualizuj ceny w magazynie docelowym
25	@id_kasy	Int	0	0 dla dokumentów magazynowych
26	@wplata_biezaca	Decimal(14,2)	0	0 dla dokumentów magazynowych
27	@zaliczka	Decimal(14,2)	0	0 dla dokumentów magazynowych
28	@id_uzytkownika	Numeric		id_uzytkownika zatwierdzającego dokument z tabeli UZYTKOWNIK
29	@id_pracownika	Numeric		id_pracownika przypisanego do użytkownika z tabeli PRACOWNIK
30	@uwagi	Varchar(500)	''	pole opisowe uwag
31	@ro_id	Numeric	0	
32	@ro_forma	Varchar(30)	''	
33	@ro_termin	Int	0	
34	@Data_VAT	Int		Data vat w postaci: round(convert(real,getdate()),0) + 36163 brak uzupełnionej zmiennej spowoduje ustawienie tej daty na datę dokumentu
35	@odlozony_mp	Tinyint	0	0 – standardowo wystawiony dokument MP 1- MP odłożony w czasie

36	@podlega_pp	Tinyint	3	0 - nie podlega MPP 1 - obowiązkowe MPP 2 - dobrowolne MPP 3 - ustal automatycznie
----	-------------	---------	---	---

Przykład wywołania procedury

```
EXEC JL_ZatwierdzDokumentMagazynowy_Server 21,1,"WZ
",78347,2,0,0,1,200,246,0,"<auto>","WZ
#####$$",1,1,1,1,1,"Netto",0,0,0,0,0,1,0,0,3000001,0,"",0,"",0,78347
```

1.4 Procedura usuwająca dokument magazynowy

Procedura i jej parametry

```
Exec JL_AnulujDokumentMagazynowy
@id_dok_mag, @id_dok_handl,
@id_firmy, @id_mag,
@id_typu, @data,
@autonumer, @akcja,
@kasuj_tylko_dokmag, @id_uzytkownika,
@kontekst
```

Opis parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@ id_dok_mag	Numeric		id_dok_magazynowego z tabeli DOKUMENT_MAGAZYNOWY
2.	@id_dok_handl	Numeric	0	0 dla dokumentów magazynowych bez dokumentów handlowych
3.	@id_firmy	Numeric		id_firmy z tabeli FIRMA
4.	@id_mag	Numeric		id_magazynu z tabeli MAGAZYN
5.	@id_typu	Numeric		id_typu z tabeli typ_dokumentu_magazynowego
6.	@data	Int		round(convert(real,getdate()),0) + 36163
7.	@autonumer	Int	0	wstawić 0 lub pole Autonumer tabeli Dokument_Magazynowy danego dokumentu
8.	@akcja	Char(3)	Del	stały tekst 'Del'
9.	@kasuj_tylko_dokmag	Tinyint		= 1 - tylko dokument magazynowy = 0 - dokument magazynowy i powiązany handlowy
10.	@id_uzytkownika	Numeric		id_uzytkownika zatwierdzającego dokument
11.	@kontekst	Tinyint	1	

1.5 Przykładowa procedura zatwierdzająca dokument magazynowy

```
-- Procedura zatwierdzająca dokument magazynowy z jedną pozycję.
-- Dla wielu pozycji należało by wykonać dodawanie pozycji w cursorze.

if exists (select 1 from sysobjects where name =
'MAGSRC_DodajDokumentMagazynowy' and type = 'P')
    drop procedure MAGSRC_DodajDokumentMagazynowy
go
create procedure MAGSRC_DodajDokumentMagazynowy
@IdFirmy numeric, @IdMagazynu numeric, @IdObiektu numeric, @IdUzytkownika
numeric, @IdWybranegoWierszaTabDod numeric, @Wynik1 varchar(255) OUTPUT,
@Wynik2 varchar(255) OUTPUT
as

-- deklaracja zmiennych
declare @errmsg varchar(255),
        @IndeksKatalogowy varchar(50),
        @IdArtykulu numeric,
        @IdJednostki numeric,
        @id_powst_dokumentu numeric,
        @id_typu numeric,
        @data int,
        @CenaBruttoZakupu decimal(16,2),
        @CenaNettoZakupu decimal(16,2),
        @KodVatZakupu varchar(5),
        @skrot_jedn varchar(5),
        @przelicznik int,

        @wartosc_pozycji_netto decimal(14,2),      -- output
        @wartosc_pozycji_brutto decimal(14,2),    -- output
        @r_id_poz_dok numeric,                    -- output
        @err_cichy varchar(255),
        @ret tinyint,

        @suma_netto decimal(16,2),
        @suma_brutto decimal(16,2),

        @num_format varchar(50),
        @okresnumeracji tinyint,
        @parametr1 tinyint,      -- podstawiamy pod @num_auto
        @parametr2 tinyint,
        @numer_zo varchar(20),
        @id_pracownika numeric

begin
    set xact_abort on
    set transaction isolation level REPEATABLE READ
    begin transaction

-- przykładowe odczytanie informacji o zmiennych dla artykułu
select @IdArtykulu = a.id_artykulu, @IdJednostki = a.ID_JEDNOSTKI,
        @CenaBruttoZakupu = CENA_ZAKUPU_BRUTTO, @CenaNettoZakupu =
cena_zakupu_netto,
        @KodVatZakupu = VAT_ZAKUPU, @skrot_jedn = j.skrot , @przelicznik =
pz.przelicznik
```

```

from ARTYKUL a inner join jednostka j
on a.id_jednostki = j.id_jednostki
    inner join pozycja_zamowienia pz
    on pz.id_artykulu = a.id_artykulu
    where isnull(a.pole1, '') = 'Część' and ID_MAGAZYNU = @IdMagazynu and
pz.id_zamowienia = @IdObiektu

--odczytanie id_typu do procedury
select @id_typu = id_typu from typ_dokumentu_magazynowego where sygnatura =
'PW' and ID_FIRMY = @Idfirmy

--konwersja daty aktualnej na clarionow1
    select @data = round(convert(real,getdate()),0) + 36163

--ustalenie pracownika
select @id_pracownika = isnull(id_pracownika,0) from pracownik where
id_uzytkownika = @IdUzytkownika

--dodanie nagłówka dokumentu magazynowego
exec AP_DodajWTleDokMag_Server
        @IdFirmy,
        NULL,
        @IdMagazynu,
        @data,
        @id_typu,
        @Wynik1,
        1,
        0,
        'PZ',
        'Z',
        'Netto',
        @IdUzytkownika,
        @id_powst_dokumentu OUTPUT

--zatwierdzenie pozycji w dokumencie
exec @ret = JL_ZatwierdzPozycje_Server
        @id_powst_dokumentu,
        NULL,
        @IdArtykulu,
        'P',
        'T',
        @skrot_jedn,
        @przelicznik,
        1,
        @KodVatZakupu,
        @CenaNettoZakupu,
        @CenaBruttoZakupu,
        0,
        0,
        0,
        2,
        0,
        2,
        'FIFO',
        1,
        @data,
        'Z',
        0,
        0,
        '',
        0,

```

```

0,
0,
0,
@wartosc_pozycji_netto OUTPUT,
@wartosc_pozycji_brutto OUTPUT,
@r_id_poz_dok OUTPUT,
0,
@err_cichy OUTPUT,
'k'
if @ret=0
    Update POZYCJA_DOKUMENTU_MAGAZYNOWEGO
    set
        ID_POZ_ORYGINALNEJ = @r_id_poz_dok,
        ID_OST_KOREKTY     = @r_id_poz_dok,
        ID_POZ_KORYGOWANEJ = @r_id_poz_dok,
        ID_POW_KOREKTY     = @r_id_poz_dok,
        FLAGA_STANU        = 0

    from pozycja_dokumentu_magazynowego p

    where p.id_poz_dok_mag = @r_id_poz_dok
else
goto error

--obliczenie sumy dokumentu
exec JL_SumujDokumentMagazynowy_Server
    @id_powst_dokumentu
    , 'PZ'
    , 'Netto'
    , 1
    , @suma_netto OUTPUT
    , @suma_brutto OUTPUT

--ustalenie numeracji
exec JL_PobierzFormatNumeracji_Server
    @IdFirmy,
    1,
    @id_typu,
    @IdMagazynu,
    @num_format OUTPUT,
    @okresnumeracji OUTPUT,
    @parametr1 OUTPUT,
    @parametr2 OUTPUT

--zatwierdzenie dokumnetu magazynowego
exec JL_ZatwierdzDokumentMagazynowy_Server
    @id_powst_dokumentu,
    @id_typu,
    'PZ',
    @data,
    @Wynik1,
    0,
    1,
    0,
    @suma_netto,
    @suma_brutto,
    NULL,
    '<auto>',
    @num_format,
    @okresnumeracji,
    @parametr1,

```



```

        @parametr2,
        @IdFirmy,
        @IdMagazynu,
        'Netto',
        NULL,
        NULL,
        0,
        0,
        0,
        0,
        0,
        0,
        @IdUzytkownika,
        @id_pracownika,
        '' ,
        0,
        '' ,
        0,
        @errmsg OUTPUT,
        @data

--przykładowe zwrócenie numeru dokumentu do @Wynik1
        select @Wynik2 = numer from dokument_magazynowy where
ID_DOK_MAGAZYNOWEGO = @id_powst_dokumentu

--przykładowy update na dokumencie
        update dbo.dokument_magazynowy set uwagi = 'Dokument wygenerowany z
zamowienia: '+ @numer_zo
        where id_dok_magazynowego = @id_powst_dokumentu

        if @@trancount>0 commit transaction
        goto Koniec
Error:
        raiserror (@errmsg,16,1)
        if @@trancount>0 rollback tran
        goto Koniec
Koniec:
        set transaction isolation level READ COMMITTED
        return
    end
go

```

2. Procedury dodające dokument handlowy

Większość danych i obsługę dokumentów realizują procedury składowane. Dla pól, które nie występują jako parametry tych procedur należy wykonać standardowe zdanie SQL:

```

Update Dokument_handlowy
Set <nazwa pola> = <wartość>
Where id_dokumentu_handlowego = <unikalny w bazie danych
identyfikator dokumentu>

```

Powyższy Update będzie stosowany najczęściej dla pól dodatkowych obsługiwanych w WAPRO Mag Prestiż i Prestiż Plus jeśli pola są wykorzystywane, jeśli nie będą wykorzystywane update nie musi zostać wykonany.

2.1 Dodanie nagłówka dokumentu handlowego

Wymaga wywołania kolejno procedur JL_DodajParagonServer_Pre, AP_DodajWTleDokMag_Server

2.1.1 Dodanie nagłówka dokumentu handlowego – inicjacja wprowadzania dokumentu

Procedura i jej parametry

EXEC JL_DodajParagonServer_Pre

@id_sesji, @id_mag,

@id_firmy, @data,

@id_kontrahenta, @id_typu,

@miejsce, @id_dok_handlowego OUTPUT, @tryb_rejestracji

Opis parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1	@id_sesji	Int		identyfikator użytkownika (pole id_uzytkownika z tabeli UŻYTKOWNIK), wprowadzającego dokument (zalogowanego do programu)
2	@id_mag	Int		identyfikator magazynu (z tabeli MAGAZYN) w kontekście, którego jest wprowadzany dokument
3	@id_firmy	Int		identyfikator firmy z tabeli FIRMA, w której rejestrowany jest dokument
4	@data	Int		data w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
5	@id_kontrahenta	Int		identyfikator kontrahenta dokumentu (z tabeli KONTRAHENT)
6	@id_typu	Int		identyfikator typu dokumentu handlowego (z tabeli TYP_DOKUMENTU_HANDLWEGO) (faktura, rachunek, paragon itp.)
7	@miejsce	Varchar(50)		miejsce wystawienia dokumentu – domyślnie inicjowane z parametru konfiguracyjnego
8	@id_dok_handl	Numeric		Zmienna OUTPUT zwracająca identyfikator nagłówka dokumentu handlowego
9	@trybrejestracji	Tinyint	0	

Przykładowe wywołanie

EXEC JL_DodajParagon_Pre 3000001,3,1,74226,1,1,'Warszawa' , @id_dok_handl OUTPUT , 0

2.1.2 Procedura JL_PobierzFormatNumeracji_Server

Procedura i jej parametry

```
EXEC JL_PobierzFormatNumeracji_Server  
    @id_firmy , @dokument,  
    @id_typu, @id_zasobu , @format_num OUTPUT,  
    @okresnumeracji OUTPUT, @parametr1 OUTPUT,  
    @parametr2 OUTPUT
```

Opis Parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Int		Id_firmy z tabeli firma
2.	@dokument	Tinyint	1	
3.	@id_typu	Int		Id_typu z tabeli typ_dokumentu_handlowego
4.	@id_zasobu	Int		Jest to id_magazynu z tabeli MAGAZYN

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury JL_ZatwierdzDokumentHandlowy_Server jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze JL_DodajWTleDokumentMagazynowy
1.	@format_numeracji	Varchar(50)	@num_format
2.	@okresnumeracji	Tinyint	@num_okresNumeracji
3.	@Parametr1	Tinyint	@num_auto
4.	@Parametr2	TinyInt	@num_niezalezny

2.1.3 Automatyczne utworzenie dokumentu magazynowego

W trakcie dodawania dokumentu handlowego musimy utworzyć nagłówek dokumentu magazynowego. Procedura ta musi być uruchomiona przed dodaniem pozycji do

dokumentu. W przypadku kiedy faktura ma mieć pozycje z kilku magazynów musi ona zostać uruchomiona oddzielnie dla pozycji powiązanych z danym magazynem.

Procedura i jej parametry

EXEC AP_DodajWTleDokMag_Server
 @id_firmy, @id_dok_handl, @id_mag,
 @data, @id_typu, @id_kontr,
 @przychod, @rozchod, @rodzaj,
 @wycena, @brutto_netto, @id_uzy, @id_dokumentu OUTPUT

Opis parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Numeric		identyfikator firmy z tabeli FIRMA, w której rejestrowany jest dokument
2.	@id_dok_handl	Numeric		identyfikator dokumentu handlowego, do którego rejestrowany jest dokument magazynowy, w przypadku korzystania z tej procedury przy dodawaniu tylko dokumentu magazynowego (bez dok. Handlowego) ustawiamy na 0
3.	@id_mag	Numeric		– identyfikator magazynu (z tabeli MAGAZYN) w kontekście, którego jest wprowadzany dokument
4.	@data	Int		w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
5.	@id_typu	Numeric		identyfikator typu dokumentu magazynowego (z tabeli TYP_DOKUMENTU_MAGAZYNOWEGO). Może to być np. WZ/PZ lub dla usług SU/ZU
6.	@id_kontr	Numeric		identyfikator kontrahenta, dla którego wystawiany jest dokument handlowy
7.	@przychod	Tinyint		flaga oznaczająca, że dokument jest przychodowy (=1) lub nie (=0)
8.	@rozchod	TinyInt		flaga oznaczająca, że dokument jest rozchodowy (=1) lub nie (=0)
9.	@rodzaj	Varchar(3)		sygnatura dokumentu magazynowego z tabeli TYP_DOKUMENTU_MAGAZYNOWEGO odpowiednia dla id_typu np. WZ
10.	@wycena	Char(1)		jednoznakowa wartość oznaczająca czy dokument jest wyceniany w cenach zakupu (=‘Z’) czy w cenach sprzedaży (=‘S’)
11.	@brutto_netto	Char(6)		sposób obliczania wartości na dokumencie = ‘Netto’ lub ‘Brutto’
12.	@id_uzy	Numeric		identyfikator użytkownika z tabeli UŻYTKOWNIK dodającego dokument

13.	@Id_dokumentu	Numeric		Zmienna OUTPUT zwracająca id zapisanego nagłówka dokumentu magazynowego
-----	---------------	---------	--	--

Procedura zwraca wartość id_dok_magazynowego z tabeli DOKUMENT_MAGAZYNOWY będącą identyfikatorem nagłówka utworzonego dokumentu magazynowego.

Przykład wywołania:

- dla dokumentów WZ

```
EXEC JL_DodajWTleDokumentMagazynowy 1,59,3,74226,1,1,0,1,'WZ ','S','Netto ',3000001,'WZ
####/$$',1,1,0,'<auto>          ',0
```

- dla dokumentów SU

```
EXEC JL_DodajWTleDokumentMagazynowy 1,59,9,74226,11,1,0,1,'WZ ','S','Netto ',3000001,'SU
####/$$',1,1,0,'<auto>          ',0
```

Oba polecenia są podobne z tym, że o prawidłowym dodaniu dokumentu decyduje id_typu.

2.1.4 Pomocnicza procedura obliczająca ilość w jednostce magazynowania dla wskazanej „innej jednostki”

Procedura i jej parametry

```
EXEC AP_PobierzJednostkeArtykulu @id_jed, @id_fir, @id_art, @skrot
```

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_jed	Numeric		Wewnętrzne id jednostek
2.	@id_fir	Numeric		identyfikator firmy z tabeli FIRMA, w której rejestrowany jest dokument
3.	@id_art	Numeric		id_artykułu z tabeli artykul
4.	@skrot	Varchar(10)		Skrót jednostki, dla której ma być obliczona ilość w jednostce magazynowania

Przykładowe wywołanie

```
EXEC AP_PobierzJednostkeArtykulu 2,1,34,'opk.'
```

2.2 Procedura zatwierdzająca pozycję dokumentu handlowego jednocześnie do dokumentu magazynowego i handlowego

Exec JL_ZatwierdzPozycje_Server
 @id_dok_mag,@id_dok_handl,@id_artykulu,@rodzaj_pozycji,@rodzaj_artykulu,@jednostka,
 @przelicznik,@ilosc,@kod_vat,@cena_netto,@cena_brutto,
 @cena_netto_wal,@cena_brutto_wal,
 @rabat,@znak_rabatu,@rabat2,@znak_rabatu2,
 @fifo_lifo,@ostrzegaj_stan_zero,@data,@wycena,
 @opakowania_wydano,@opakowania_przyjeto,@pozycja_drs
 @nr_paczki,@Oblicz_wartosc_pozycji,@tryb_rejestracji,@id_poz_dostawy,
 @wartosc_pozycji_netto,@wartosc_pozycji_brutto,@r_id_poz_dok,@tryb_cichy,
 @err_cichy,@zn_ceny,@czb_marza

Opis Parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_dok_mag	Numeric		Wewnętrzne pole dokumentu (pole ma właściwość IDENTITY) zwracane przez JL_DodajDokumentMagazynowy
2.	@id_dok_handl	Numeric		Id powiązanego dokumentu handlowego, najlepiej wstawić wartość zmiennej @Id_dok_handlowego procedury dodającej nagłówki dokumentu handlowego.
3.	@id_art	Numeric		id_artykulu na pozycję z tabeli artykul
4.	@rodzaj_poz	Char(1)		'R' dla rozchodu lub 'P' dla przychodu
5.	@rodzaj_art	Char(1)		Pierwszy znak z pola rodzaj z tabeli artykul np. T- Towar, P-Produkt, U- Usługa, itd.
6.	@jednostka	Varchar(10)		skrót jednostki z tabeli jednostka
7.	@przelicznik	Decimal(16,6)		przelicznik jednostki z tabeli jednostka
8.	@ilosc	Decimal(16,6)		ilość asortymentu na pozycji w jednostce magazynowania
9.	@kod_vat	Char(3)		kod stawki VAT z tabeli stawka_vat
10.	@cena_netto	Decimal(14,2)		Cena netto

11.	@cena_brutto	Decimal(14,2)		Cena brutto
12.	@cena_netto_wal	Decimal(14,4)	0 – dla dokumentów magazynowych	Cena netto w walucie
13.	@cena_brutto_wal	Decimal(14,4)	0 – dla dokumentów magazynowych	Cena brutto w walucie
14.	@rabat	Decimal(8,2)		Procent rabatu lub narzutu
15.	@znak	Tinyint		znak rabat 1 kwoty dla parametru - @rabatu >=0 wartość 2 w przeciwnym wypadku 0
16.	@rabat2	Decimal(8,2)	0 – dla dokumentów magazynowych	Procent rabatu z dokumentu handlowego
17.	@znak2	Tinyint		znak rabat 2 kwoty dla parametru - @rabatu2 >=0 wartość 2 w przeciwnym wypadku 0
18.	@fifo_lifo	Char(4)		wartości 'FIFO' lub 'LIFO'
19.	@ostrzegaj_stan_zero	Tinyint		0 nie ostrzegaj lub 1 ostrzegaj
20.	@data	Int		round(convert(real.getdate()),0) + 36163
21.	@wycena	Char(1)		Z' wg cen zakupu lub 'S' wg cen sprzedaży
22.	@op_wydano	Decimal(14,4)	0 – jeśli towar nie jest opakowaniem	Pola używane dla asortymentu opakowanie
23.	@op_przyjeto	Decimal(14,4)	0 – jeśli towar nie jest opakowaniem	Pola używane dla asortymentu opakowanie
24.	@pozycja_drs	Tinyint	0	
25.	@nr_paczki	Varchar(20)	“	opis klasy towaru wymagany dla faktur RR – bez znaczenia w pozostałych przypadkach opis klasy towaru wymagany dla faktur RR – bez znaczenia w pozostałych przypadkach
26.	@Oblicz_wartosc_pozycji	Tinyint	0	Czy obliczyć wartość pozycji 0 – nie , 1 – tak, uzupełniane są wtedy zmienne w punkcie 29 oraz 30
27.	@tryb_rejestracji	Tinyint	0	Pole wewnętrzne.
28.	@id_poz_dostawy	Numeric	0	jeżeli rozchód jest robiony z dostawy to należy przekazać id_poz_dok_mag z wybranej pozycji przychodowej

29.	@wartosc_pozycji_netto	Decimal(14,2)		Parametr OUTPUT Wartość netto pozycji
30.	@wartosc_pozycji_brutto	Decimal(14,2)		Parametr OUTPUT Wartość brutto pozycji
31.	@r_id_poz_dok	Numeric(18)		Parametr OUTPUT – zwraca id_poz_dok_mag zapisanej pozycji
32.	@tryb_cichy	tinyint	0	Jeśli 0 to brak komunikatu jeśli 1 to zwrócona wartości do @err_cichy
33.	@err_cichy	Tinyint		Błąd zwrócony podczas zapisywania pozycji - parametr OUTPUT
34.	@zn_ceny	Char(1)	k	k kartotekowa p promocyjna i indywidualna g dla grupy cenowej m modyfikowana ręcznie
35.	@czb_marza	Decimal(14,4)	0	>0 gdy zatwierdzamy pozycje dokumentu rozchodu podlegającego VAT marża
36.	@nr_serii_we	Varchar(50)	''	numer serii preferowanej dostawy z której ma nastąpić rozchód. Parametr używany przy ustawieniu w kartotece asortymentowej metody rozchodu wg numeru serii/daty ważności
37.	@Data_waznosci_we	Int	0	data ważności preferowanej dostawy z której ma nastąpić rozchód (analogicznie jak powyżej)

2.3 Procedura zatwierdzająca dokument handlowy

Zatwierdzenie dokumentu powinno się odbyć poprzez wywołanie kolejno procedur JL_PobierzFormatNumeracji_Server, JL_ZatwierdzDokumentHandlowy_Server, AP_GenerujRozliczenieDokumentuHandlowego

2.3.1 Sumowanie dokumentu handlowego procedurą JL_SumujDokumentHandlowy_Server

Procedura i jej parametry

```
EXEC JL_SumujDokumentHandlowy_Server
    @id_dok_handl ,
    @rodzaj_dok,
    @Brutto_netto,
    @nr_filtra,
    @Suma_netto OUTPUT,
    @Suma_brutto OUTPUT,
    @Suma_netto_wal OUTPUT,
    @Suma_brutto_wal OUTPUT
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
-----	-----------------	------------	------------------	--------------------

5	@id_dok_handl	Int		Id dokumentu handlowego z tabeli dokument handlowy lub zwrócony parametr z procedury JL_DodajParagonServer_Pre
6	@rodzaj_dok	Char(2)		Dla zakupu= 'z' dla sprzedaży='s'
7	@brutto_netto	Char(6)		Ewidencja dokumentu 'Netto' lub 'Brutto'
8	@nr_filtra	Int	1	

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury JL_ZatwierdzDokumenthandlowy_Server jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze JL_ZatwierdzDokumenthandlowy_Server
1.	@suma_netto	Decimal(16,6)	@suma_netto
2.	@suma_brutto	Decimal(16,6)	@suma_brutto
3.	@suma_netto_wal	Decimal(16,2)	@suma_netto_wal
4.	@suma_brutto_wal	Decimal(16,2)	@suma_brutto_wal

2.3.2 Procedura JL_PobierzFormatNumeracji_Server

Procedura i jej parametry

```
EXEC JL_PobierzFormatNumeracji_Server
  @id_firmy ,
  @dokument,
  @id_typu,
  @id_zasobu ,
  @format_num OUTPUT,
  @okresnumeracji OUTPUT,
  @parametr1 OUTPUT,
  @parametr2 OUTPUT
```

Opis Parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
-----	-----------------	------------	------------------	--------------------

1.	@id_firmy	Int		Id_firmy z tabeli firma
2.	@dokument	Tinyint	4	
3.	@id_typu	Int		Id_typu z tabeli typ_dokumentu_handlowego
4.	@id_zasobu	Int		Jest to id_magazynu z tabeli MAGAZYN

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury JL_ZatwierdzDokumentHandlowy jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze JL_ZatwierdzDokumentHandlowy
1.	@format_numeracji	Varchar(50)	@num_format
2.	@okresnumeracji	Tinyint	@num_okresNumeracji
3.	@Parametr1	Tinyint	@num_auto
4.	@Parametr2	TinyInt	@num_niezalezny

2.3.3 Procedura zatwierdzania dokumentu handlowego

Procedura i jej parametry

EXEC JL_ZatwierdzDokumentHandlowy_Server

@id_dok_handl, @id_typu, @numer,
 @num_format, @num_okresnumeracji, @num_auto
 @num_niezalezna, @id_firmy, @id_mag, @forma_platnosci,
 @miejsce, @data_wplywu, @data_wystawienia,
 @data_sprzedazy, @id_kontrahenta, @id_platnika,
 @odebral, @uwagi, @rabat,
 @znak, @termin_plat, @zaliczka,
 @wplata_biezaca, @wplata_biezaca_wal, @wartosc_netto,
 @znak_wn, @wartosc_brutto, @dok_wal,
 @sym_wal, @przelicznik_wal, @data_kursu_wal,

@wartosc_netto_wal, @wartosc_brutto_wal, @rodzaj_dok,
 @brutto_netto, @fifo_lifo, @przeliczaj_ceny,
 @id_dok_korygowanego, @fiskalny_ok, @id_kasy,
 @id_uzytkownika, @id_pracownika, @tryb_rejestracji,
 @id_rachunku, @potwierdzony_ue,@trojstronny_ue,
 @id_fzal_org, @bez_errmsg

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_dok_handl	Numeric		identyfikator dokumentu handlowego, zwracany przez procedurę JL_DodajParagonServer_Pre
2.	@id_typu	Numeric		id_typu z tabeli TYP_DOKUMENTU_HANDLOWEGO
3.	@numer	Varchar(30)		numer dokumentu handlowego. Wartość = '<auto>' w trybie automatycznej numeracji
4.	@num_format	Varchar(30)		maska formatu numeracji
5.	@num_okresnumeracji	Tinyint		ciągły lub miesięczny lub roczny, wartości (1,0,2,3)
6.	@num_auto	tinyint		numeracja automatyczna = 1 lub ręczna 2
7.	@num_niezalezna	Tinyint		numeracja zależna od magazynu lub nie(1- zależna , 2 -niezależna)
8.	@id_firmy	Int		identyfikator firmy, w której rejestrowany jest dokument
9.	@id_mag	Int		identyfikator magazynu w kontekście, którego rejestrowany jest dokument
10.	@forma_platnosci	Varchar(20)		nazwa formy płatności dokumentu (pole NAZWA z tabeli FORMA_PLATNOSCI)
11.	@miejsce	Varchar(50)		miejsce wystawienia dokumentu
12.	@data_wplywu	Int		data wpływu dokumentu = round(convert(real,getdate()),0) + 36163
13.	@data_wystawienia	Int		data wystawienia (rejestracji) = round(convert(real,getdate()),0) + 36163
14.	@data_sprzedazy	Int		data sprzedaży lub zakupu w zależności od typu dokumentu = round(convert(real,getdate()),0) + 36163
15.	@id_kontrahenta	Tinyint		identyfikator kontrahenta (ID_KONTARHENTA z tabeli KONTRAHENT)
16.	@id_platnika	Tinyint		identyfikator płatnika (pole ID_PLATNIKA z tabeli KONTRAHENT) – nabywcy towaru

17.	@odebral	Varchar(50)	"	informacja kto, odebrał towar (pole 50 znaków maks.)
18.	@uwagi	Varchar(1000)	"	uwagi do dokumentu (pole 500 znaków maks.),
19.	@rabat	Decimal(9,2)		procent rabatu narzutu do całego dokumentu
20.	@znak	Tinyint		– znak kwoty dla parametru @rabat (@rabatu >=0 wartość 2 w przeciwnym wypadku 0
21.	@termin_plat	Int		termin płatności dokumentu – data w formacie wewnętrznym = np.: round(convert(real,getdate()),0) + 36163
22.	@zaliczka	Decimal(14,2)		kwota zaliczki pobranej przy rejestracji dokumentu
23.	@wplata_biezaca	Decimal(14,2)		wartość wpłaty bieżącej zarejestrowana przy wystawianiu dokumentu
24.	@wplata_biezaca_wal	Decimal(14,2)		wartość wpłaty bieżącej w walucie obcej dokumentu
25.	@wartosc_netto	Decimal(14,2)		suma wartości netto dokumentu
26.	@znak_wn	Tinyint		znak kwoty dla parametru @wartosc_netto (@wartosc_netto >=0 wartość 2 w przeciwnym wypadku 0
27.	@wartosc_brutto	Decimal(14,2)		suma wartości brutto dokumentu
28.	@dok_wal	Tinyint	0	flaga czy dokument w walucie obcej (=1) czy też nie (=0)
29.	@sym_wal	Char(3)	"	symbol waluty jeśli @dok_wal=1 w przeciwnym razie puste
30.	@przelicznik_wal	Decimal(14,8)	1	kurs waluty jeśli dokumentnie nie jest walutowy to 1
31.	@data_kursu_wal	Int	0	data wprowadzonego w parametrze @przelicznik_wal kursu waluty jeśli nie jest dokument walutowy to 0
32.	@wartosc_netto_wal	Decimal(14,2)	0	suma wartości netto dokumentu w walucie obcej (lub 0)
33.	@wartosc_brutto_wal	Decimal(14,2)	0	suma wartości brutto dokumentu w walucie obcej (lub 0)
34.	@rodzaj_dok	Char(1)		jednoznakowa wartość = 's' jeśli dokument sprzedaży i ='z' jeśli zakupu
35.	@brutto_netto	Char(6)		sposób obliczania dokumentu – wartość 'Netto' lub 'Brutto'
36.	@fifo_lifo	Char(4)		sposób zdejmowania ze stanu magazynu = 'FIFO' lub 'LIFO'
37.	@przeliczaj_ceny	tinyint	0	sposób przeliczania cen w kartotece asortymentowej = 0 (nie przliczaj), =1 (tylko ceny zakupu), =2 (ceny zakupu i sprzedaży wg istniejącej marży)
38.	@id_dok_korygowane go	Numeric	0	identyfikator dokumentu handlowego – korygowanego wystawianym dokumentem, =0 jeśli dokument pierwotny

39.	@fiskalny_ok	Tinyint	0	flaga informująca czy dokument został ufiskalniony (=1) czy nie (=0)
40.	@id_kasy	Int		identyfikator kasy ID_KASY z tabeli KASY_FIRMY, w której użytkownik rejestruje dokument handlowy
41.	@id_uzytkownika	Numeric		identyfikator użytkownika rejestrującego dokument handlowy z tabeli użytkownik
42.	@id_pracownika	Numeric		identyfikator pracownika wystawiającego dokument (z tabeli PRACOWNIK)
43.	@tryb_rejestracji	Tinyint		tryb rejestracji dokumentu – normalny dokument obrotowy (=0), dokument nie zdejmujący ze stanu magazynu (=1)
44.	@id_rachunku	Numeric		identyfikator rachunku bankowego - dla dokumentów sprzedaży (@rodzaj_dok='s') jest to ID_RACHUNKU z tabeli RACHUNEK_FIRMY dla dokumentów zakupu (@rodzaj_dok='z') jest to ID_RACHUNKU z tabeli RACHUNEK_KONTRAHENTA
45.	@potwierdzony_ue	Tinyint	1	
46.	@trojstronny_ue	Tinyint	1	
47.	@id_fzal_org	Numeric	0	
48.	@bez_errmsg	Tinyint	0	1 jeśli komunikat błędu z procedury ma być pominięty
49.	@uproszczony	Tinyint	0	1 jeśli dokument uproszczony
50.	@Zaliczka_odroczonea	Tinyint	0	1 w przypadku faktur zaliczkowych z zaliczką odroczoną
51.	@Data_otrzymania	Int		Data otrzymania – data w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163 Ważna przy dokumentach zakupu
52.	@kraj_ewidencji_vat	Varchar(3)	'PL' lub ''	
53.	@numer_zewnetrzny	Varchar(100)	''	Data otrzymania to data otrzymania faktury. Numer zewnętrzny to numer wystawcy faktury istotny o ile ktoś ma automatyczną numerację faktur zakupu
54.	@podlega_pp	Tinyint	3	0 - nie podlega MPP 1 - obowiązkowe MPP 2 - dobrowolne MPP 3 - ustal automatycznie

Przykład wywołania:

```
EXEC JL_ZatwierdzDokumentHandlowy 59,1,'<auto>
','F/#####/$$',1,1,2,1,3,'przelew','Warszawa',74226,74226,74226,1,1,',',',',0,2,74233,0,0,0,1231.56,2,1
502.5,0,',',',',0,0,0,0,'s','Netto ',',',0,0,0,1,3000001,0,0,3,1,1,0
```

2.3.4 Procedura generująca dokument finansowy do zatwierdzonej faktury AP_GenerujRozliczenieDokumentuHandlowego

Do wersji 8.00.8 odpowiedzialna za generację dokumentu była procedura zatwierdzająca fakturę. W momencie wprowadzenia funkcji obsługi terminali płatniczych musiało to zostać zmienione i za tą obsługę odpowiada procedura AP_GenerujRozliczenieDokumentuHandlowego.

Procedura ta powinna być wywołana po zatwierdzeniu dokumentu handlowego.

Procedura i jej parametry

```
EXEC AP_GenerujRozliczenieDokumentuHandlowego
@id_firmy, @id_mag, @id_dok_handl,
@wplata_biezaca @sym_wal, @zaliczka,
@id_kasy, @id_rachunku, @id_uzytkownika,
@id_pracownika, @znak_wn
```

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	numeric		identyfikator firmy, w której rejestrowany jest dokument
2	@id_mag	Numeric		identyfikator magazynu w kontekście, którego rejestrowany jest dokument
3.	,@id_dok_handl	Numeric		identyfikator dokumentu handlowego, zwracany przez procedurę JL_DodajParagonServer_Pre
4.	@wplata_biezaca	Decimal(16,2)		wartość wpłaty bieżącej zarejestrowana przy wystawianiu dokumentu
5.	@sym_wal	Varchar(3)		symbol waluty jeśli @dok_wal=1 w przeciwnym razie puste
6.	@zaliczka	Decimal(14,2)		kwota zaliczki pobranej przy rejestracji dokumentu
7.	@id_kasy	Int		identyfikator kasy ID_KASY z tabeli KASY_FIRMY, w której użytkownik rejestruje dokument handlowy
8.	@id_rachunku	Numeric		identyfikator rachunku bankowego - dla dokumentów sprzedaży (@rodzaj_dok='s') jest to ID_RACHUNKU z tabeli RACHUNEK_FIRMY dla dokumentów zakupu (@rodzaj_dok='z') jest to ID_RACHUNKU z tabeli

				RACHUNEK_KONTRAHENTA
9.	@id_uzytkownika	Numeric		identyfikator użytkownika rejestrującego dokument handlowy z tabeli użytkownik
10.	@id_pracownika	Numeric		identyfikator pracownika wystawiającego dokument (z tabeli PRACOWNIK)
11.	@znak_wn	Tinyint		znak kwoty dla parametru @wartosc_netto (@wartosc_netto >=0 wartość 2 w przeciwnym wypadku 0

Przykład wywołania procedury

EXEC AP_GenerujRozliczenieDokumentuHandlowego 1,1,22,123,"",0,1,0,3000001,0,2

2.4 Wyjątki – dla procedury zatwierdzającej dokument handlowy – korekta

2.4.1 Procedura dodająca dokument handlowy korekty

Procedura i jej parametry

EXEC JL_DodajDokumentKorekty
 @id_dok_korygowanego, @id_dok_oryginalnego, @data,
 @id_kontrahenta,@id_platnika, @id_firmy,
 @id_magazynu, @id_typu, @id_sesji

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_dok_korygowanego	Numeric		identyfikator id dokumentu handlowego z tabeli DOKUEMNT_HANDLOWY, który jest korygowany (dla pierwszej korekty @id_dok_korygowanego = @id_dok_oryginalnego)
2.	@id_dok_oryginalnego	Numeric		- identyfikator id dokumentu handlowego z tabeli DOKUEMNT_HANDLOWY - dokumentu pierwotnego
3.	@data	Int		data rejestracji korekty = round(convert(real,getdate()),0) + 36163
4.	@id_kontrahenta	Numeric		Należy pobrać z dokumentu korygowanego
5.	@id_platnika	Numeric		Należy pobrać z dokumentu korygowanego
6.	@id_firmy	Numeric		Należy pobrać z dokumentu korygowanego

7.	@id_magazynu	Numeric		Należy pobrać z dokumentu korygowanego
8.	@id_typu	Numeric		identyfikator typu dokumentu korekty z tabeli TYP_DOKUMENTU_HANDLOWEGO
9.	@id_sesji	Numeric		identyfikator użytkownika z tabeli UŻYTKOWNIK
10.	@Data_otrzymania	Int		data otrzymania korekty = round(convert(real,getdate()),0) + 36163
11.	@Korektakontrahenta	Tinyint	0	parametr określa czy dodawany dokument korekty dotyczy jedynie zmiany kontrahenta (korekty) na dokumencie handlowym (1- tak , 0 – nie)

Przykład wywołania:

EXEC JL_DodajDokumentKorekty 19,19,74226,10,10,1,1,3,3000001

2.4.2 Korygowanie pozycji

Procedura i jej parametry

EXEC JL_ZatwierdzZmianePozycji

@id_dok_mag, @id_dok_handl, @id_poz_dok, @id_art, @rodzaj_poz,
@rodzaj_art, @data, @jednostka, @przelicznik, @kodVAT,
@cena_netto, @cena_brutto, @cena_netto_wal, @cena_brutto_wal,@rabat,
@znak, @op_wydano, @op_przyjeto, @ilosc_new, @ilosc_old,
@fifo_lifo,@wycena, @akcja, @id_poz_korygowanej, @nr_paczki,
@tryb_rejestracji

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_dok_mag	Numeric		identyfikator dokumentu magazynowego, dla którego rejestrowana jest pozycja
2.	@id_dok_handl	Numeric		identyfikator dokumentu handlowego, dla którego rejestrowana jest pozycja
3.	@id_poz_dok,	Numeric		Identyfikator pozycji oryginalnej
4.	@id_art	Numeric		identyfikator artykułu (pole id_artykułu z tabeli ARTYKUL)
5.	@rodzaj_poz	Char(1)		jednoznakowa wartość oznaczająca, że pozycja jest rozchodowa (= 'R') lub przychodowa (= 'P')
6.	@rodzaj_art	Char(1)		jednoznakowa wartość będąca pierwszą literą rodzaju artykułu z tabeli ARTYKUL np. T- Towar, P-Produkt, U- Usługa
7	@Data	Int		round(convert(real,getdate()),0) + 36163 data rejestracji pozycji

8.	@jednostka	Varchar(10)		skrót jednostki artykułu z tabeli jednostka,
9.	@przelicznik	Decimal(16,6)		przelicznik jednostki z tabeli jednostka,
10.	@kod_vat	Char(3)		kod stawki VAT z tabeli STAWKA_VAT
11.	@cena_netto,	Decimal(14,4)		Cena netto pozycji
12.	@cena_brutto	Decimal(14,4)		Cena brutto pozycji
13.	@cena_netto_wal	Decimal(14,4)	0	tylko dla faktur w walucie – w pozostałych przypadkach =0,
14.	@cena_brutto_wal	Decimal(14,4)	0	tylko dla faktur w walucie – w pozostałych przypadkach =0
15.	@rabat	Decimal(8,2)		procent rabatu lub narzutu
16.	@znak	Tinyint		znak wartości dla parametru @rabat (@rabatu >=0 wartość 2 w przeciwnym wypadku 0
17.	@op_wydano	Decimal(14,4)	0	poła używane dla asortymentu Opakowanie – jeśli pozycja jest opakowaniem
18.	@op_przyjeto	Decimal(14,4)	0	poła używane dla asortymentu Opakowanie
19.	@ilosc_new	Decimal(16,4)		nowa ilość asortymentu na pozycji <u>w jednostce magazynowania</u>
20.	@ilosc_old	Decimal(16,4)		poprzednia ilość asortymentu na pozycji <u>w jednostce magazynowania</u>
21.	@fifo_lifo	Char(4)		wartości 'FIFO' lub 'LIFO' – sposób zdejmowania ze stanu magazynu
22.	@wycena	Char(1)		= 'Z' wg cen zakupu lub = 'S' wg cen sprzedaży
23.	@akcja	Char(3)		w przypadku korekt = 'Upd'
24.	@id_poz_korygowanej	Numeric	0	identyfikator ID_POZ_DOK_MAG z tabeli POZYCJA_DOKUMENTU_MAGAZYNOWEGO, która jest korygowana
25.	@nr_paczki	Varchar(20)	"	opis klasy towaru wymagany dla faktur RR – bez znaczenia w pozostałych przypadkach
26.	@tryb_rejestracji	Tinyint	0	tryb rejestracji dokumentu – normalny dokument obrotowy (=0), dokument nie zdejmujący ze stanu magazynu (=1)
27.	@zn_ceny	Char(1)	0	k kartotekowa p promocyjna i indywidualna g dla grupy cenowej m modyfikowana ręcznie
28.	@czb_marza	Decimal(14,4)	0	>0 gdy zatwierdzamy pozycje dokumentu rozchodu podlegającego VAT marża
29.	@nr_serii_we	Varchar(50)	"	numer serii preferowanej dostawy z której ma nastąpić rozchód. Parametr używany przy ustawieniu w kartotece asortymentowej metody

				rozchodu wg numeru serii/daty ważności
30.	@data_waznosci_we	Int	0	data ważności preferowanej dostawy z której ma nastąpić rozchód (analogicznie jak powyżej)

Przykładowe wywołanie:

```
EXEC JL_ZatwierdzZmianePozycji 15,61,285,21,'R','T',0,'szt.',1,'22',164.96,201.25,0,0,0,2,0,0,1,2,'FIFO','','Upd',112,"",0,0,0
```

2.4.3 Procedura do zatwierdzania dokumentu korekty

Procedura jest taka sama jak do zatwierdzenia dokumentu handlowego - JL_ZatwierdzDokumentHandlowy – punkt 2.3.3.

2.4.4 Procedura rozliczająca dokument korekty.

Do wersji 8.00.8 odpowiedzialna za generację dokumentu była procedura zatwierdzająca fakturę korektę. Z powodu wprowadzenia funkcji obsługi terminali płatniczych musiało to zostać zmienione i za tą obsługę odpowiada procedura AP_GenerujRozliczenieDokumentuHandlowego.

Wywołanie procedury jest takie same jak dla dokumentu faktury - punkt 2.3.4.

2.5 Przykład dodania dokumentu handlowego FZUE

Poniższa procedura nie jest kompletna ale prezentuje kolejność wykonywania procedur do prawidłowego zatwierdzenia dokumentu handlowego zakupu.

```
if exists (select 1 from sysobjects where name = 'MAGEXT_ImportFZUEXML_H' and type = 'P')
drop procedure MAGEXT_ImportFZUEXML_H
go
create procedure dbo.MAGEXT_ImportFZUEXML_H
@id_firmy numeric,
@id_magazynu numeric,
@id_uzytkownika numeric
as

declare @errmsg varchar(500),
@id_typuPZ int,
@id_typuZU int,
@id_typuFZ int,
@IdArtykulu numeric,
@skrot_jedn varchar(10),
@data int ,
@id_pracownika int ,
@id_powst_dokumentuPZ int ,
```

@id_powst_dokumentuZU int ,
@id_powst_dokumentu int,
@ret int,
@przelicznik numeric(16,4),
@wartosc_pozycji_netto numeric (16,2),
@wartosc_pozycji_brutto numeric (16,2),
@r_id_poz_dok numeric ,
@err_cichy varchar(255),
@suma_netto numeric(16,2),
@suma_brutto numeric(16,2),
@num_format varchar(50),
@okresnumeracji tinyint,
@num_auto tinyint, -- podstawiamy pod @num_auto
@num_niezalezna tinyint,
@NumerFZ varchar(30),
@IdKontr numeric,
@IdKontr_Platnik numeric,
@NIP varchar(30),
@Rodzaj varchar(5),
@id_dok_handl numeric,
@miejscowosc varchar(40),
@paymentType varchar(20),
@forma_platnosci varchar(20),
@termin_zobowiazan int,
@suma_netto_wal decimal(16,2),
@suma_brutto_wal decimal(16,2),
@Data_TXT varchar(20),
@Data_DT datetime,
@Data_Sales_TXT varchar(20),
@Data_Sales_DT datetime,
@Data_Sales int,
@dateDue_TXT varchar(20),
@dateDue_DT datetime,
@id_kasy int,
@BruttoNetto varchar(10),
@id_kontr numeric ,
@IdGrupaKontr numeric,
@IdKlasyfKontrah numeric ,
@DomyslnaFormaPlatZkonfFirmy varchar(50),
@dateTax_TXT varchar(20),
@dateTax_DT datetime,
@dateTax int,
@quantity numeric(16,4),
@discountPercentage numeric(16,4),
@code varchar(100),
@unitPrice numeric(16,4),
@priceVAT varchar(10),
@unitGrossPrice numeric(16,4),
@unitPricePLN numeric(16,4),
@unitGrossPricePLN numeric(16,4),
@waluta varchar(10),
@Kurs numeric(16,6),
@I_MASA_NETTO numeric(16,3),
@kod_cn varchar(10),
@KOD_TRANSPORTU varchar(10),
@KOD_TRANSAKCJI varchar(10),
@KOD_DOSTAWY varchar(10),
@I_KOD_KRAJU_POCH varchar(10),
@Id_Kosztu numeric ,
@CenaNettoKoszt numeric(16,2),

```
@CenaBruttoKoszt numeric(16,2)
```

```
begin
```

```
select @BruttoNetto = 'Netto'
```

```
--odczytanie typów dokumentów
```

```
select @id_typuPZ = id_typu from dbo.typ_dokumentu_magazynowego with (nolock) where  
sygnatura = 'PZ' and id_firmy = @Id_Firmy
```

```
select @id_typuZU = id_typu from dbo.typ_dokumentu_magazynowego with (nolock) where  
sygnatura = 'ZU' and id_firmy = @Id_Firmy
```

```
select @id_typuFZ = id_typu from dbo.TYP_DOKUMENTU_HANDLOWEGO with (nolock) where  
sygnatura = 'FZUE' and id_firmy = @Id_Firmy
```

```
--przykładowe daty do dokumentu
```

```
select @data = round(cast(@Data_DT as real),0,1)+36163,  
@termin_zobowiazan = round(cast(@dateDue_DT as real),0,1)+36163,  
@dateTax = round(cast(@dateTax_DT as real),0,1)+36163
```

```
select @Data_Sales = @Data,  
@NumerFZ = 'FZ/8278383/06'
```

```
--odszukanie kontrahenta
```

```
select @IdKontr = id_kontrahenta,  
@IdKontr_Platnik = id_platnika  
from dbo.KONTRAHENT I with (nolock) where NIP = @NIP and ID_FIRMY = @Id_Firmy
```

```
--odszukanie pracownika
```

```
if not exists( select 1 from dbo.pracownik p with (nolock) where id_uzytkownika = @Id_Uzytkownika  
and ID_FIRMY = @Id_Firmy )  
select @id_pracownika = 0  
else  
select @id_pracownika = isnull(id_pracownika,0) from pracownik with (nolock) where id_uzytkownika  
= @Id_Uzytkownika and ID_FIRMY = @Id_Firmy
```

```
--sprawdzenie czy numer dokumentu jest unikatowy
```

```
declare @RetStat int,  
@rok int,  
@rodzaj_dok char(1)
```

```
set @rodzaj_dok = 'z'
```

```
select @rok=YEAR(@Data_DT)
```

```
Exec @RetStat= RM_SprawdzUnikalnoscNumeru @id_firmy, @id_magazynu,  
@NumerFZ,4,@id_typuFZ,1,@IdKontr,@rok  
if @RetStat=0 and exists(select 1 from DOKUMENT_HANDLOWY with (nolock) where numer =  
@NumerFZ  
and id_kontrahenta=case when @rodzaj_dok<>'z' then id_kontrahenta  
else @IdKontr end  
and TYP_DOKUMENTU=@rodzaj_dok)
```

```
begin
```

```
select @errmsg = 'Numer dokumentu importowanego występuje już w bazie danych. Operacja  
przerwana.'
```

```
GOTO error  
end
```

```
set xact_abort on
```

```
set transaction isolation level REPEATABLE READ
begin transaction
```

```
--odczytanie z konfiguracji firmy miejsca wystawienia
```

```
exec AP_PobierzParamKonf 's',@id_firmy,'MiejWystFak/Rach',@miejscowosc output
```

```
--dodanie dokumentu handlowego
```

```
EXEC dbo.JL_DodajParagonServer_Pre
```

```
@id_uzytkownika,
```

```
@id_magazynu,
```

```
@id_firmy,
```

```
@data,
```

```
@IdKontr,
```

```
@id_typuFZ,
```

```
@miejscowosc,
```

```
@id_dok_handl OUTPUT,
```

```
0
```

```
--dodatkowy update
```

```
UPDATE dbo.dokument_handlowy SET typ_dokumentu='z',
      MP=0,
      POTWIERDZONY_UE=1,
      TROJSTRONNY_UE=0,
      dok_korekty=null,
      numer=@NumerFZ, --'FZ/TEST/14', --
      obliczanie_wg_cen=@BruttoNetto, --'Netto',
      dok_wal=1, --1,
      id_platnika=@IdKontr_Platnik,
      data_sprzedazy=@Data_Sales,
      forma_platnosci='przelew',
      termin_plat=@termin_zobowiazan,
      odebral="",
      id_rachunku=0,
      data_wplywu=@data,
      trybrejestracji=0,
      DATA_KURS_WAL=@DataKurs,
      SYM_WAL=@Waluta,
      PRZELICZNIK_WAL=@Kurs,
      DOK_WAL_KRAJOWY=0,
      NARZUT_RABAT=0,
      ID_PRACOWNIKA=0,
      ID_FZAL_ORG=0,
      ID_FZAL_OST=0,
      UWAGI="",
      NUMER_PARAGONU="",
      PRZELICZNIK_WAL_PZ = @Kurs,
      DATA_KURS_WAL_PZ=@DataKurs,
      WYROZNIK='<auto>',
      GRUPUJ_POZYCJE=0,
      WEWNETRZNY=1,
      MARZOWY=0,
      ID_BANKU=0,
      POLE1="",
      POLE2="",
      POLE3="",
      POLE4= "",
      POLE5= "",
      POLE6= "",
      POLE7= "",
      POLE8= "",
```

```

        POLE9= ",
        POLE10= ",
        ODWROTNY=1
    WHERE id_dokumentu_handlowego=@id_dok_handl;

```

--założenie dokumentów magazynowych do dokumentu handlowego

```

exec AP_DodajWTleDokMag_Server
    @Id_Firmy,
    @id_dok_handl,
    @Id_Magazynu,
    @Data_Sales,
    @id_typuPZ,
    @IdKontr,
    1,
    0,
    'PZ',
    'Z',
    @BruttoNetto, --'Netto',
    @Id_Uzytkownika,
    @id_powst_dokumentuPZ OUTPUT

```

```

exec AP_DodajWTleDokMag_Server
    @Id_Firmy,
    @id_dok_handl,
    @Id_Magazynu,
    @Data_Sales,
    @id_typuZU,
    @IdKontr,
    1,
    0,
    'PZ',
    'Z',
    @BruttoNetto, --'Netto',
    @Id_Uzytkownika,
    @id_powst_dokumentuZU OUTPUT

```

--odczytanie i dodanie pozycji do dokumentu pozycji

```

DECLARE Odczyt_PozycjiArt CURSOR LOCAL FAST_FORWARD FOR

```

```

select p.quantity,p.discountPercentage, p.code, p.unitPrice, p.priceVAT
from #TMP_PozPZ p
where len(p.code)<=20

```

```

OPEN Odczyt_PozycjiArt
WHILE 1=1
BEGIN
    FETCH NEXT FROM Odczyt_PozycjiArt INTO
        @quantity,
        @discountPercentage,
        @code,
        @unitPrice,
        @priceVAT

```

```

    IF @@fetch_status<>0 BREAK

```

```

set @IdArtykulu = 0

```

```
select @IdArtykulu = a.id_artykulu from dbo.artykul a with (nolock)
where a.ID_MAGAZYNU = @Id_magazynu and a.INDEKS_KATALOGOWY = @code
```

SELECT

```
@Rodzaj = upper(substring(a.rodzaj,1,1)),
@skrot_jedn = j.skrot,
@przelicznik = j.przelicznik,
@unitGrossPrice = dbo.JL_Func_ObliczBrutto(@unitPrice.@priceVAT),
@unitPricePLN = 0 ,
@unitGrossPricePLN = 0 ,
@I_MASA_NETTO = round(@quantity * isnull(a.WAGA,0),3),
@I_KOD_KRAJU_POCH = isnull(a.KRAJ_POCHODZENIA,"")
```

```
FROM dbo.artykul a WITH(NOLOCK) LEFT JOIN dbo.jednostka j WITH(NOLOCK)
```

```
ON a.id_jednostki=j.id_jednostki
```

```
LEFT JOIN dbo.stawka_vat sv WITH(NOLOCK)
```

```
ON a.vat_zakupu=sv.KOD_VAT
```

```
WHERE a.ID_ARTYKULU = @IdArtykulu
```

```
select @id_powst dokumentu = case when @Rodzaj = 'U' then @id_powst dokumentuZU else
@id_powst dokumentuPZ end
```

```
EXEC @ret = dbo.JL_ZatwierdzPozycje_Server
```

```
@id_powst dokumentu,
@id_dok_handl, --null,
@IdArtykulu,
'P',
@Rodzaj,
@skrot_jedn,
@przelicznik,
@quantity, --@ilosc,
@priceVAT, --@KodVatZakupu,
@unitPricePLN, --pln
@unitGrossPricePLN, --pln
@unitPrice,
@unitGrossPrice,
0, --rabat
2, --znak rabatu
0, --rabat 2
2, --znak rabat 2
'FIFO',
1,
@Data_Sales,
'z', --wycena
0, --op wyd
0, --op prz
0, --poz drs
", --nr_paczki
0, --obl wartosc
0, --tryb rej
0, --id_poz_dostawy
@wartosc_pozycji_netto OUTPUT,
@wartosc_pozycji_brutto OUTPUT,
@r_id_poz_dok OUTPUT,
0,
@err_cichy OUTPUT,
'k',
0
```

```
if @ret=0 or @ret = 2
```

```
Update POZYCJA_DOKUMENTU_MAGAZYNOWEGO
```

```
set --OPIS = p.opis,
```

```

NR_SERII          = "",
DATA_WAZNOSCI    = 0, --", --@data_waznosci_sys,
ID_POZ_ORYGINALNEJ = @r_id_poz_dok,
ID_OST_KOREKTY   = @r_id_poz_dok,
ID_POZ_KORYGOWANEJ = @r_id_poz_dok,
ID_POW_KOREKTY   = @r_id_poz_dok,
FLAGA_STANU      = 0,
OPIS_CENY = "",
POLE1            = ",--p.POLE1,
POLE2            = ",--p.POLE2,
POLE3            = ",--p.POLE3,
POLE4            = ",--p.POLE4,
POLE5            = ",--p.POLE5,
POLE6            = ",--p.POLE6,
POLE7            = ",--p.POLE7,
POLE8            = ",--p.POLE8,
POLE9            = ",--p.POLE9,
POLE10           = "--p.POLE10,
--CN_SPRZ_PZ     = p.CN_SPRZ_PZ,
--CB_SPRZ_PZ     = p.CB_SPRZ_PZ

```

```

from pozycja_dokumentu_magazynowego p1
where p1.id_poz_dok_mag = @r_id_poz_dok
else
BEGIN
select @errmsg = @err_cichy
GOTO Error
END

```

--doczytanie danych na potrzeby intrastatu

```

set @kod_cn = ""
set @KOD_DOSTAWY = ""
set @KOD_TRANSPORTU = ""
set @KOD_TRANSAKCJI = ""

```

```

select @kod_cn = isnull(kod_cn, "") from ARTYKUL a with (nolock) where ID_ARTYKULU =
@IdArtykulu

```

```

select @KOD_DOSTAWY = KOD_DOSTAWY from INTRA_K_DOSTAWY with (nolock) where
domyslny=1

```

```

select @KOD_TRANSPORTU = KOD_TRANSPORTU from INTRA_K_TRANSPORTU with (nolock)
where domyslny=1

```

```

select @KOD_TRANSAKCJI = KOD_TRANSAKCJI from INTRA_K_TRANSAKCJI with (nolock)
where domyslny=1

```

```

EXEC AP_WyliczCenyBazowe @id_dok_handl,@r_id_poz_dok,@BruttoNetto,@Kurs,0

```

Update POZYCJA_DOKUMENTU_MAGAZYNOWEGO

```

set
I_PODLEGA_DEKLARACJI=1,
I_KOD_CN =@kod_cn,
I_KOD_TRANSPORTU =@KOD_TRANSPORTU,
I_KOD_TRANSAKCJI =@KOD_TRANSAKCJI,
I_KOD_DOSTAWY =@KOD_DOSTAWY,
I_KOD_KRAJU_POCH =@I_KOD_KRAJU_POCH,
I_MASA_NETTO=@I_MASA_NETTO
where ID_POZ_DOK_MAG = @r_id_poz_dok

```



```

END
CLOSE Odczyt_PozycjiArt
DEALLOCATE Odczyt_PozycjiArt

--ustalenie numeracji
EXEC dbo.JL_PobierzFormatNumeracji_Server @id_firmy,
    4,
    @id_typuFZ,
    @id_magazynu,
    @num_format OUTPUT,
    @okresnumeracji OUTPUT,
    @num_auto OUTPUT,
    @num_niezalezna OUTPUT

SELECT @forma_platnosci = 'przelew'

--odczytanie danych o kasie z konfiguracji użytkownika
SELECT @id_kasy = case when isnumeric(WARTOSC) = 1 then convert(int, WARTOSC) else 0 end
FROM dbo.INTERUZYTKOWNIKPOZYCJA i WITH (NOLOCK) inner join POZYCJAKONFIG pk with
(nolock)
on i.ID_POZYCJI = pk.ID_POZYCJI
WHERE id_firmy = @ID_Firmy and ID_Uzytkownika = @id_uzytkownika and pk.KOD = 'DomKasa'

--dodanie kosztów

set @Id_Kosztu = 0

--odczytanie kosztów z tabelki dodatkowej
if exists (select 1 from #TMP_PozPZ where code ='Transport' and [unitPrice] > 0 )
begin --początek kosztów

set @priceVAT = ""

select top 1 @CenaNettoKoszt = unitPrice,
    @CenaBruttoKoszt = dbo.JL_Func_ObliczBrutto(unitPrice,priceVAT),
    @priceVAT = i.priceVAT
from #TMP_PozPZ i
where code ='Transport' and [unitPrice] > 0

if exists (select 1 from KOSZT_DOK_HANDL)
begin
    select @Id_Kosztu = max(ID_KOSZTU)+1 from KOSZT_DOK_HANDL
end
else
begin
    select @Id_Kosztu = 1
end

INSERT INTO dbo.KOSZT_DOK_HANDL
    (ID_KOSZTU,
    ID_KOSZTU_KOR,
    ID_DOKUMENTU_HANDLOWEGO,
    NAZWA,
    ILOSC,
    CENA_NETTO,
    CENA_BRUTTO,

```

```

KOD_VAT,
WARTOSC_NETTO,
WARTOSC_BRUTTO,RODZAJ_KOSZTU,BEZ_WPLYWU_NA_VAT,
ID_KONTRAHENTA,
ID_ROZRACHUNKU,NUMER,TERMIN_PLAT,FORMA_PLAT,DATA_WYST_FAK,DATA_OT
RZ_FAK,DATA_OPER_GOSP,ROZRACH_NB)
VALUES (@ID_kosztu,0,@id_dok_handl,'Transport',1,@CenaNettoKoszt,@CenaBruttoKoszt,
@priceVAT,
@CenaNettoKoszt, --wart net
@CenaBruttoKoszt --wart brutto
,2 --RODZAJ_KOSZTU
,1 --BEZ_WPLYWU_NA_VAT
,0 --ID_KONTRAHENTA
,0 --ID_ROZRACHUNKU
"--NUMER
,0, --TERMIN_PLAT
"--FORMA_PLAT
0, --DATA_WYST_FAK
0, --DATA_OTRZ_FAK
0, --DATA_OPER_GOSP
0 --ROZRACH_NB
)

```

```

insert into inter_koszt_poz (id_kosztu,id_poz_dok_mag)
select @Id_Kosztu,id_poz_dok_mag
from pozycja_dokumentu_magazynowego p with (nolock)
where p.id_dok_handlowego=@id_dok_handl
and id_poz_dok_mag not in (select id_poz_dok_mag from inter_koszt_poz where
id_kosztu=1)
and p.RODZAJ_ARTYKULU <>'U'

```

```
EXEC dbo.AP_WyliczCenyBazowe @id_dok_handl,0,'Netto',@Kurs,0
```

```
end --koniec kosztów.
```

```
--dane do rejestrów
```

```
EXEC RM_UstalMomentyVAT
@id_firmy,@id_dok_handl,@data,@dateTax,'FZUE','z',@waluta,'PLN',@data,@dateTax
```

```
EXEC JL_ObliczVATDokumentuHandlowego @id_dok_handl,0,@BruttoNetto
```

```
EXEC RM_DoRejestru_ModyfikacjaVAT @id_dok_handl
```

```
--dla FUE jest inaczej
```

```
select @suma_netto = netto,
@suma_brutto = brutto,
@suma_netto_wal = netto_wal,
@suma_brutto_wal = brutto_wal
from dbo.AP_Func_PodajVatDokHandl(@id_dok_handl)
```

```
--zatwierdzenie dokumentu handlowego
```

```
EXEC dbo.JL_ZatwierdzDokumentHandlowy_Server
@id_dok_handl,
@id_typuFZ,
@NumerFZ,
@num_format,
```

```

@okresnumeracji,
@num_auto,
@num_niezalezna,
@id_firmy,
@id_magazynu,
@forma_platnosci,
@miejscowosc,
@Data_Sales, --data wplywy -- data przyjecia towaru
@data, --data wysta -- data wystawienia
@data, --data sprz --data zakupu
@IdKontr,
@IdKontr_Platnik ,
"
"
,
0,
2,
@termin_zobowiazan,
0,
0,
0,
@suma_netto,
2,
@suma_brutto,
1, --1, dok_walutowy
@waluta,
@Kurs,
@DataKurs,
@suma_netto_wal,
@suma_brutto_wal,
'Z',
@BruttoNetto, --'Netto',
'FIFO',
0,
0,
0,
@id_kasy,
@id_uzytkownika,
@Id_Pracownika,
0, --tryb rej
0, --id_rach
1, --potwi ue
0, --trojst
0, --idfzalog
0, --bez errmsg
0, --upr
0, -- zal odr
@dateTax -- data otrzymania

```

```
EXEC RM_AktualizujDokumentHandlowy @id_dok_handl,0,0,0,",@DataKurs,1,0,0,1,"
```

```
--obsługa rozliczenia
```

```
declare @WplataBierzaca numeric(16,4),
```

```
@Zaliczka numeric (16,4)
```

```
select @WplataBierzaca = 0
```

```
,@Zaliczka = 0
```

```
EXEC AP_GenerujRozliczenieDokumentuHandlowego
```

```
@Id_Firmy,
```

```
@id_magazynu,
```

```
@id_dok_handl,
```

```

@WplataBierzaca,
@waluta,
@Zaliczka,
@id_kasy,
0, --idRachunku
@id_uzytkownika,
@ID_Pracownika,
2 --znak_wn

--ustalenie pól dodatkowych
update Dokument_handlowy
set pole1="",pole2="",pole3="",pole4="",pole5="",pole6="",pole7="",pole8="",pole9="",pole10=""
where id_dokumentu_handlowego=@id_dok_handl

--zapis do loga
update dokument_handlowy set uwagi = left('Dokument zakupu zaimportowano z pliku XML.',1000)
where id_dokumentu_handlowego = @id_dok_handl

declare @komunikatLog varchar(100)

select @komunikatLog = left('Zaimportowany dokument z pliku: '+cast(ltrim(rtrim(@NumerFZ)) as
varchar(20)),100)

exec AP_ZapiszDoLogu
@id_firmy,@id_magazynu,@id_uzytkownika,@id_dok_handl,'HAN_CSV',1,@komunikatLog,3

if @@trancount>0 commit transaction
goto Koniec
Error:
raiserror (@errmsg,16,1)
if @@trancount>0 rollback tran
goto Koniec
Koniec:
set transaction isolation level READ COMMITTED
return
end
GO

```

3. Dodanie dokumentu finansowego

Większość danych i obsługę dokumentów realizują procedury składowane. Dla pól, które nie występują jako parametry tych procedur należy wykonać standardowe zdanie SQL:

```

Update Dokument_finansowy
Set <nazwa pola> = <wartość>
Where id_dokumentu_finansowego = <unikalny w bazie danych
identyfikator dokumentu>

```

Powyższy Update będzie stosowany najczęściej dla pól dodatkowych obsługiwanych w WAPRO Mag Prestiż i Prestiż Plus jeśli pola są wykorzystywane, jeśli nie będą wykorzystywane update nie musi zostać wykonany.

3.1 Procedura dodająca nagłówek dokumentu finansowego

Procedura i jej parametry

```
exec AP_DodajDokumentFinansowy_Server @idfir,
```

@id_typu, @id_plat, @typ_plat , @iduzyt,
@id_dok_fin **output**

Opis parametrów.

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru/Opis
1.	@ idfir	Numeric		identyfikator firmy z tabeli FIRMA
2.	@id_typu	Numeric		identyfikator typu dokumentu (z tabeli typ_dokumentu_finansowego)
3.	@id_plat	Numeric		Identyfikator kontrahenta z tabeli kontrahent
4.	@typ_plat	Char(1)		Małe litery -> k – kontrahent, p-pracownik, u –urząd
5	@id_uzytkownika	Numeric		
6.	@id_dok_fin	Numeric OUTPUT		zwrotnie w @id_dok_fin dostajemy identyfikator dokumentu finansowego

3.2 Procedura AP_DodajPozycjeDlaDokFin

Procedura i jej parametry

Tej procedury używamy jeśli pozycje dokumentu finansowego mają być dodane bez rozliczania rozrachunku. Ważne wtedy jest ustawienie 6-tego parametru @cz_r na 0.

```
EXEC AP_DodajPozycjeDlaDokFin @idf, @kw, @kw_wal,  
@wal_bazowa, @opis, @czy_r, @kier, @rodzaj
```

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru/Opis
1.	@idf	Numeric		identyfikator dokumentu_finansowego
2.	@kw	Decimal(16,4)		Kwota złotówkowa pozycji
3.	@kw_wal	Decimal(16,4)		Kwota walutowa pozycji
4.	@wal_bazowa	Tinyint	0	jeśli KP jest w walucie to ustawiamy 1 – rozrachunek powstaje nam rozrachunek w walucie bazowej
5.	@opis	Varchar(100)		Opis pozycji dodawanej
6.	@czy_r	Tinyint	0	0 – standardowa pozycja dok. Fin 1-

				zaliczka 2 - Rozliczenie innego rozrach.
7.	@kier	Char(1)		W – wpływ, Y-wypływ
8.	@rodzaj	Char(1)	" – pusta wartość	lub P w pozycji generuj płatność

3.3 Procedura AP_WciagnijRozrachunkiNaPozycje

Procedura i jej parametry

Tej procedury używamy do dodawania pozycji dokumentu finansowego która ma rozliczać rozrachunki.

exec AP_WciagnijRozrachunkiNaPozycje @idf, @iduzyt, @sym_wal

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru/Opis
1.	@idf	Numeric		identyfikator dokumentu_financego
2.	@iduzyt	Numeric		identyfikator użytkownika
3.	@sym_wal	Varchar(3)	"	Puste jeśli dokument nie jest walutowy

Przed uruchomieniem tej procedury należy wpisać do tabeli zaznaczone identyfikatory rozrachunków do "id", 5 – do kolumny "użycie", identyfikator użytkownika z tabeli użytkownik do kolumny "id_sesji"

Np.:

```
Insert into zaznaczone (Id_Sesji, uzycie, id , ilość, id2 ) select
(@id_uzytkownika, 5, @id_rozrachunku, 0,0)
```

3.4 Zatwierdzanie dokumentu finansowego

Do poprawnego zatwierdzenia dokumentu zalecane jest uruchomienie kolejno procedur JL_PobierzFormatNumeracji_Server, AP_ZatwierdzDokFin.

3.4.1 Procedura JL_PobierzFormatNumeracji_Server

Procedura i jej parametry

EXEC JL_PobierzFormatNumeracji_Server

@id_firmy ,
 @dokument,
 @id_typu,
 @id_zasobu ,
 @format_num OUTPUT,
 @okresnumeracji OUTPUT,
 @parametr1 OUTPUT,
 @parametr2 OUTPUT

Opis Parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Int		Id_firmy z tabeli firma
2.	@dokument	Tinyint	3	
3.	@id_typu	Int		Id_typu z tabeli typ_dokumentu_finansowego
4.	@id_zasobu	Int		Jest to id_magazynu z tabeli MAGAZYN

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury AP_ZatwierdzDokFin jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze AP_ZatwierdzDokFin
1.	@format_numeracji	Varchar(50)	@num_format
2.	@okresnumeracji	Tinyint	@num_okresNumeracji
3.	@Parametr1	Tinyint	@num_auto
4.	@Parametr2	TinyInt	@num_niezalezny

3.4.2 Procedura sprawdzająca rozrachunki do rozliczenia

Procedura przydatna do sprawdzenia czy rozrachunki rozliczane można rozliczyć. Najlepiej ją dodać przed dodaniem pozycji na dokument finansowy.

Procedura i jej parametry

EXEC AP_CzyRozliczycRozrachunki @id_uzyt,@sym_wal output,@err output,@id_plat,@typ_plat

Opis Parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@iduzyt	Numeric(18)		Id_uzytkownika
2.	@sym_wal	Varchar(3) OUTPUT		Komunikat o symbol waluty
3.	@err	Varchar(100) OUTPUT		Komunikat o błędzie gdy rozrachunek nie jest rozliczony
4.	@IdPlat	Numeric(18)		id płatnika z tabeli kontrahent
5.	@typ_plat	Char(1)		Typ kartoteki k – kontrahent , p – pracownik

Przykład wywołania

EXEC AP_CzyRozliczycRozrachunki @id_uzyt,@sym_wal output,@err output,@id_plat,@typ_plat

3.4.3 Procedura zatwierdzająca dokument finansowy

Procedura i jej parametry

EXEC AP_ZatwierdzDokFin

@num_auto @num_niezalezny, @id_dok_fin, @id_typu, @data, @id_plat,

@typ_plat, @kw, @id_firmy,

@id_rach_kas, @num_okresnumeracji, @num_format ,

@nr_rach_plat, @nr_dok_fin, @nr_wyc,

@tyt , @id_prac, @uwagi,

@sym_wal, @kw_wal, @przelicznik_wal

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru/Opis
1	@num_auto	Tinyint		
2	@num_niezalezny	Tinyint		
3.	@id_dok_fin	Numeric		identyfikator dokumentu
4.	@id_typu	Numeric		identyfikator typu dokumentu (z tabeli typ_dokumentu_finansowego)
5	@Data	Int		round(convert(real.getdate()),0) + 36163
6.	@id_plat	Numeric		identyfikator kontrahenta z tabeli kontrahent
7.	@typ_plat	Char(1)		Małe litery -> k - kontrahent, p - pracownik, u - urząd
8.	@kw	Decimal(16,2)		kwota dokumentu
9.	@id_firmy	Numeric		identyfikator firmy z tabela firma

10	@id_rach_kas	Numeric		identyfikator kasy kasy_firmy
11	@numeracja	Tinyint		
12	@maska	Varchar(20)		
13	@nr_rach_plat	Varchar(100)	"	Jeśli jest to dokument kasowy to pusty string jeśli dokument bankowy to wpisujemy numer rachunku płatnika- jest to pole numer_rachunku z tabeli rachunek kontrahenta
14.	@nr_dok_fin	Varchar(30)	"	pusty string w przypadku numeracji automatycznej, numer nadany w przypadku numeracji ręcznej
15.	@nr_wyc	Varchar(50)	"	pusty string w przypadku dokumentów kasowych i rozliczeniu "inne"
16.	@tyt	Varchar(200)		tytuł dokumentu
17.	@id_prac	Numeric		identyfikator pracownika z tabeli pracownika
18.	@uwagi	Varchar(500)	"	Opis do dokumentu
19.	@sym_wal	Varchar(3)	"	pusty string w przypadku dokumentów złotówkowych lub symbol waluty,
20	@kw_wal	Decimal(16,4)		kwota walutowa, w przypadku dokumentów złotówkowych = @kw
21.	@przelicznik_wal	Decimal(14,6)	1	- kurs waluty, w przypadku dokumentów złotówkowych =1
22.	@kod_opearcji	Varchar(5)	"	Na potrzeby fakira
23.	@Poz_wal_bazowe	Tinyint	0	Dopuszczalne wartości 0 dla oraz 1 dla przypadku kp walutowego, które generuje rozrachunek złotówkowy
24.	@Id_Magazynu	Numeric	0	
25.	@Data_kurs_wal	Int	0	Data kursu
26.	@pp	Tinyint	0	parametr określa czy dokument finansowy dotyczy podzielonej płatności (1- tak , 0 – nie)
27.	@pp_NIP	Varchar(30)	"	parametr określa numer NIP podawany do przelewu do banku w komunikacie Elixir-0 podzielonej płatności
28.	@pp_nr_faktury	Varchar(30)	"	analogicznie jak Nip ale dla numeru faktury
29	@pp_kwota_vat_k	Decimal(16,2)	0	kwota podawana w przelewie split payment (przy wystawianiu dokumentu przelewu własnego)
30	@pp_kwota_vat_r	Decimal(16,2)	0	kwota faktyczna która przyszła w komunikacie przelewu obcego (nie używane w obecnej wersji)

3.5 Przykłady dodawania dokumentów finansowych.

a) Zatwierdzanie dokumentu finansowego z rozliczeniem rozrachunku

Przykładowa deklaracja zmiennych

```
DECLARE @id_firmy NUMERIC
DECLARE @id_typu NUMERIC
DECLARE @id_platnika NUMERIC
DECLARE @id_uzytkownika NUMERIC
DECLARE @id_dok_fin NUMERIC
DECLARE @num_format VARCHAR(50)
DECLARE @okresnumeracji TINYINT
DECLARE @num_auto TINYINT
DECLARE @num_niezalezny TINYINT
DECLARE @kwota DECIMAL(16,4)
DECLARE @idrozrach int
```

```
SET @id_firmy=1
SET @id_typu=4
SET @id_platnika=11
SET @id_uzytkownika=3000001
SET @kwota=234.68
SET @idrozrach = 198
```

Wywołanie procedury dodającej nagłówek dokumentu

```
EXEC AP_DodajDokumentFinansowy_Server
@id_firmy,@id_typu,@id_platnika,'k',@id_uzytkownika,@id_dok_fin OUTPUT
```

Zablokowanie rozrachunków na potrzeby rozliczenia

```
update rozrachunek_mag_v set semafor = @id_uzytkownika, flaga_stanu = 3
where id_rozrachunku = @idrozrach
```

Uzupełnienie tabeli zaznaczone

```
insert into zaznaczone VALUES (@id_uzytkownika, 5 , @idrozrach, 0,0)
```

Procedura wciągająca pozycje do dokumentu finansowego

```
EXEC AP_WciagnijRozrachunkiNaPozycje @id_dok_fin,@id_uzytkownika, ''
```

Aktualizacja kwoty na pozycji do rozliczenia. Domyślnie program zaproponuje rozliczenie całości rozrachunku, więc jeśli zawsze rozliczamy całość to UPDATE ten nie jest potrzebny

```
update inter_fin_rozrach set kwota = @kwota where id_dokumentu_finansowego
= @id_dok_fin and id_rozrachunku = @idrozrach
```

Pobranie numeracji

```
EXEC JL_PobierzFormatNumeracji_Server @id_firmy,3,@id_typu,0,@num_format
OUTPUT,@okresnumeracji OUTPUT,@num_auto OUTPUT,@num_niezalezny OUTPUT
```

Zatwierdzanie dokumentu finansowego

```
EXEC AP_ZatwierdzDokFin
@num_auto,@num_niezalezny,@id_dok_fin,@id_typu,76960,@id_platnika,'k',@kwota,
1,1,@okresnumeracji,@num_format,'10101238-
23453452345345',' ',' ','tytułem',0,' ',' ',0,1,'MPT',0
```

Wyczyszczenie tabeli zaznaczone

```
delete from zaznaczone where id = @idrozrach
```

b) Zatwierdzanie dokumentu finansowego bez rozliczania rozrachunku

Przykładowa deklaracja zmiennych

```
DECLARE @id_firmy NUMERIC
DECLARE @id_typu NUMERIC
DECLARE @id_platnika NUMERIC
DECLARE @id_uzytkownika NUMERIC
DECLARE @id_dok_fin NUMERIC
DECLARE @num_format VARCHAR(50)
DECLARE @okresnumeracji TINYINT
DECLARE @num_auto TINYINT
DECLARE @num_niezalezny TINYINT
DECLARE @kwota DECIMAL(16,4)
DECLARE @idrozrach int
```

```
SET @id_firmy=1
SET @id_typu=4
SET @id_platnika=11
SET @id_uzytkownika=3000001
SET @kwota=234.68
SET @idrozrach = 198
```

Wywołanie procedury dodającej nagłówek dokumentu

```
EXEC AP_DodajDokumentFinansowy_Server
@id_firmy,@id_typu,@id_platnika,'k',@id_uzytkownika,@id_dok_fin OUTPUT
```

Wywołanie procedury dodającej pozycje na dokument finansowy

```
EXEC AP_DodajPozycjeDlaDokFin @id_dok_fin,@kwota,0,0,'FV/00012/12',0,'W',''
```

Pobranie numeracji

```
EXEC JL_PobierzFormatNumeracji_Server @id_firmy,3,@id_typu,0,@num_format
OUTPUT,@okresnumeracji OUTPUT,@num_auto OUTPUT,@num_niezalezny OUTPUT
```

Zatwierdzanie dokumentu finansowego

```
EXEC AP_ZatwierdzDokFin
@num_auto,@num_niezalezny,@id_dok_fin,@id_typu,76960,@id_platnika,'k',@kwota,1,1,@okresnumeracji,@num_format,'10101238-23453452345345',' ',' ','tytułem',0,' ',' ',0,1,'MPT',0
```

4. Dodawanie zamówień

Większość danych i obsługę dokumentów realizują procedury składowane. Dla pól, które nie występują jako parametry tych procedur należy wykonać standardowe zdanie SQL:

```
Update zamowienie
Set <nazwa pola> = <wartość>
Where id_zamowienia = <unikalny w bazie danych identyfikator dokumentu>
```

Powyższy Update będzie stosowany najczęściej dla pól dodatkowych obsługiwanych w WAPRO Mag Prestiż i Prestiż Plus jeśli pola są wykorzystywane, jeśli nie będą wykorzystywane update nie musi zostać wykonany.

4.1 Procedura dodająca nagłówek zamówienia

Procedura i jej parametry :

EXEC RM_DodajZamowienie_Server
@id_firmy, @id_kontrahenta, @id_magazynu,
@typ, @data, @semafor
@trybrejestracji @brutto_netto, @Flaga_stanu , @Id_zamowienia OUTPUT

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru/Opis
1.	@id_firmy	Numeric		identyfikator firmy (z tabeli FIRMA), w której dodawane jest zamówienie
2.	@id_kontrahenta	Numeric		identyfikator kontrahenta (z tabeli KONTRAHENT), którego dotyczy zamówienie
3.	@id_magazynu	Numeric		identyfikator magazynu (z tabeli MAGAZYN) , w którym powstaje zamówienie
4.	@typ	Tinyint		typ zamówienia 1- od odbiorcy 2 – do dostawcy
5.	@data	Int		data utworzenia / przyjęcia zamówienia np.: round(convert(real,getdate()),0) + 36163
6.	@semafor	Numerci		identyfikator użytkownika (pole id_uzytkownika z tabeli UZYTKOWNIKA), który wprowadza zamówienie
7.	@trybrejestracji	Tinyint	0	przyjmuje dwie wartości 0 – oznacza zamówienie rejestrowane w głównym rejestrze zaś 10 – w buforze zamówień
8.	@brutto_netto	Char(6)	Netto	sposób obliczania wartości na dokumencie = 'Netto' lub ='Brutto' domyślnie 'Netto' (nie trzeba podawać parametru dla trybu Netto)
9.	@Flaga_stanu	Tinyint	1	1 – dodawania zamówienia
10.	@id_zamowienia	Numeric(18,0) OUTPUT		Zmienna OUTPUT wracająca id dodanego nagłówka zamówienia

4.2 Procedura dodawania pozycji zamówienia

Procedura i jej parametry :

Exec RM_DodajPozycjeZamowienia_Server
@id_pozycji_zamowienia, @id_zamowienia, @id_artykulu,
@kod_vat, @zamowiono, @zrealizowano,

@zarezerwowano, @do_rezerwacji, @cena_netto,
 @cena_brutto, @cena_netto_wal, @cena_brutto_wal,
 @przelicznik, @jednostka, @narzut,
 @opis, @znak_narzutu, @trybrejestracji, @id_dostawy_rez, @id_wariantu_produkту

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru/Opis
1.	@id_pozycji_zamowienia	Numeric	0	0 – dodaj nową pozycję zamówienia
2.	@id_zamowienia	Numeric		Identyfikator zamówienia zwrócony z procedury dodającej nagłówek @id_zamowienia
3.	@id_artykulu	Numeric		identyfikator artykułu z tabeli ARTYKUL , którego dotyczy pozycja
4.	@kod_vat	Varchar(3)		kod stawki VAT pozycji
5.	@zamowiono	Decimal(16,6)		ilość zamowiona towaru
	@zrealizowano	Decimal(16,6)	0	ilość zrealizowana =0,
6.	@zarezerwowano	Decimal(16,6)		Ilość zarezerwowana
7.	@do_rezerwacji	Decimal(16,6)		- ilość do rezerwacji towaru < ZAMOWIONO – ZREALIZOWANO
8.	@cena_netto	Decimal(14,4)		Cena netto na pozycji zamówienia
9.	@cena_brutto	Decimal(14,4)		Cena brutto na pozycji zamówienia
10.	@cena_netto_wal	Decimal(14,4)		= 0 dla zamówień złotych lub cena w walucie zamówienia
11.	@cena_brutto_wal	Decimal(14,4)		= 0 dla zamówień złotych lub cena w walucie zamówienia
12.	@przelicznik	Decimal(16,6)		przelicznik jednostki towaru z tabeli jednostka względem jednostki magazynowania,
13.	@jednostka	Varchar(10)		– kod jednostki
14.	@narzut	Decimal(8,4)		Wartość narzutu /rabatu
15.	@opis	Varchar(500)	”	opis pozycji
16.	@znak_narzutu	Tinyint		znak narzutu (+) = 0 lub rabatu (-) = 2
17.	@trybrejestracji	Tinyint		przyjmuje dwie wartości 0 – oznacza pozycję zamówienia rejestrowaną w głównym rejestrze zaś 10 – w buforze zamówień. Wartość odpowiada kolumnie TRYBREJESTRACJI tabeli ZAMOWIENIE
18.	@id_dostawy_rez	Numeric	0	Id_poz_dok_mag – dostawa, z której nastąpi rezerwacja

19.	@id_wariantu_produkту	Numeric	0	Wskazanie na identyfikator wariantu produktu
20.	@znacznik_ceny	Char(1)	m	k kartotekowa p promocyjna i indywidualna g dla grupy cenowej m modyfikowana ręcznie
21.	@nr_serii	Varchar(50)	"	numer serii (lub fragment początkowy numeru serii) preferowanej dostawy z jakiej ma nastąpić rezerwacja. Parametr używany przy ustawieniu w kartotece asortymentowej metody rozchodu wg numeru serii/daty ważności

4.3 Zatwierdzanie zamówienia

Operacja zatwierdzania zamówienia powinna się odbyć poprzez wywołanie kolejno procedur JL_PobierzFormatNumeracji_Server, RM_ZatwierdzZamowienie , RM_SumujZamowienie_Server

4.3.1 Procedura sumująca zamówienie

Procedura i jej parametry

```
EXEC RM_SumujZamowienie_Server
    @id_zamowienia ,
    @Suma_netto OUTPUT,
    @Suma_brutto OUTPUT,
    @Suma_netto_wal OUTPUT,
    @Suma_brutto_wal OUTPUT
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_zamowienia	Int		Id_zamowienia z tabeli zamówienie

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT):

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze RM_SumujZamowienie_Server
5.	@suma_netto	Decimal(16,6)	@suma_netto

6.	@suma_brutto	Decimal(16,6)	@suma_brutto
7.	@suma_netto_wal	Decimal(16,2)	@suma_netto_wal
8.	@suma_brutto_wal	Decimal(16,2)	@suma_brutto_wal

4.3.2 Procedura JL_PobierzFormatNumeracji_Server

Procedura i jej parametry

```
EXEC JL_PobierzFormatNumeracji_Server
  @id_firmy ,
  @dokument,
  @id_typu,
  @id_zasobu ,
  @format_num OUTPUT,
  @okresnumeracji OUTPUT,
  @parametr1 OUTPUT,
  @parametr2 OUTPUT
```

Opis Parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Int		Id_firmy z tabeli firma
2.	@dokument	Tinyint	2	
3.	@id_typu	Int		Id typu z tabeli typ_dokumentu_magazynowego dla rodzaju = 'Zamówienie'
4.	@id_zasobu	Int		Jest to id_magazynu z tabeli MAGAZYN

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury RM_ZatwierdzZamowienie_Server jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze RM_ZatwierdzZamowienie_Server
1.	@format_numeracji	Varchar(50)	@num_format
2.	@okresnumeracji	Tinyint	@num_okresNumeracji

3.	@Parametr1	Tinyint	@num_auto
4.	@Parametr2	TinyInt	@num_niezalezny

4.3.3 Procedura zatwierdzająca zamówienie

Procedura i jej parametry

EXEC RM_ZatwierdzZamowienie_Server

@id_zamowienia, @id_kontrahenta, @id_typu,
 @numer, @num_format, @num_okresnumeracji,
 @num_auto, @num_niezalezny, @autonumer,
 @id_firmy, @id_mag, @data,
 @data_realizacji, @zaliczka, @priorytet,
 @auto_rezerwacja, @nr_zam_klienta, @typ,
 @id_pracownika, @przelicznik_wal, @data_kurs_wal,
 @sym_wal, @dok_wal, @flaga_stanu,
 @rabat_narzut, @znak_rabatu, @uwagi, @InformacjeDodatkowe, @Osoba zamawiająca,
 @Id_kontakt, @FormaPlatnosci, @DniPlatnosci

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_zamowienia,	Numeric		wewnętrzne id dokumentu
2.	@id_kontrahenta	Numeric		Identyfikator kontrahenta z tabeli KONTRAHENT
3.	@id_typu	Numeric		- identyfikator typu dokumentu (pole id_typu z tabeli TYP_DOKUMENTU_MAGAZYNOWEGO
4.	@numer	Varchar(30)		numer dokumentu handlowego. Wartość ='<auto> ' w trybie automatycznej numeracji,
5.	@num_format	Varchar(30)		maska formatu numeracji
6.	@num_okresnumeracji	tinyint		ciągły ,miesięczny ,roczny, dzienny (1,0,2,3)
7.	@num_auto	Tinyint		numeracja automatyczna = 1 lub ręczna 2
8.	@num_niezalezny	Tinyint		numeracja zależna od magazynu lub nie(1- zależna , 2 -niezależna)
9.	@Autonumer	Int		
10.	@id_firmy	Numeric		identyfikator firmy, w której rejestrowany jest dokument

11.	@id_mag	Numeric		identyfikator magazynu w kontekście, którego rejestrowany jest dokument
12.	@data	Int		data wystawienia zamówienia w formacie wewnętrznym = round(convert(real.getdate()),0) + 36163
13.	@data_realizacji	Int		– jw. – termin realizacji zamówienia
14.	@zaliczka	Decimal(14,4)		suma zaliczek dodanych podczas wystawiania zamówienia
15	@priorytet	Tinyint		priorytet zamówienia od 1 do 3 (1 najważniejsze ,3 – najmniej ważne)
16.	@auto_rezerwacja	Tinyint		czy automatycznie towar ma być rezerwowany do wielkości ilości zamówiono – zrealizowano , 1- tak , 0 - nie
17.	@nr_zam_klienta	Varchar(30)	''	nr zamówienia przesłanego przez klienta
18.	@typ	Tinyint		=1 zamówienie od odbiorcy, zamówienie do dostawcy = 2
19	@id_pracownika	Numeric		pole z tabeli PRACOWNIK – identyfikator pracownika wystawiającego / odpowiedzialnego za obsługę zamówienia
20	@przelicznik_wal	Decimal(16,4)	1	wartość przelicznika waluty w przypadku zamówień walutowych, przy złotówkowych zamówieniach = 1
21	@data_kurs_wal	Int		data wprowadzonego do pola przelicznik_wal kursu waluty
22	@sym_wal	Varchar(3)	''	symbol waluty, puste dla PLN,
23	@dok_wal	Tinyint		= 1 dokument walutowy,=0 dokument złotówkowy (w walucie bazowej)
24.	@flaga_stanu	Tinyint		– wskaźnik trybu zatwierdzania zamówienia =1 dodawanie, = 2 edycja
25.	@trybrejestracji	Tinyint		– przyjmuje dwie wartości 0 – oznacza zamówienie rejestrowane w głównym rejestrze zaś 10 – w buforze zamówień
26.	@rabat_narzut	Decimal(6,2)		wielkość rabatu lub narzutu wyrażona w procentach np. 5 (w przykładzie) oznacza 5%
27.	@znak_rabatu	Tinyint		zmienna określająca czy jest to rabat (na minus = 0) lub narzut (czyli na plus =2). Wartości te przyjęte są z uwagi na sposób interpretacji – wartość rabatu/narzutu w procedurze ustalana jest za pomocą wzoru: (@znak_rabatu-1)*@rabat_narzut
28.	@uwagi	Varchar(1000)		uwagi tekstowe do złożonego zamówienia
29.	@InformacjeDodatkowe	Varchar(1000)	''	Pole informacje dodatkowe

30	@Osoba zamawiająca	Varchar(101)	"	Imię i nazwisko w tabeli kontakt
31	@Id_kontaktu	Numeric	0	Identyfikator kontaktu z tabeli KONTAKT
32	@FormaPlatnosci	Varchar(50)	"	Nazwa formy płatności z tabeli FORMY_PLATNOSCI
33	@DniPlatnosci	Int	0	Ilość dni dla formy płatności
34	@FakturaParagon	Tinyint	0	Określamy tutaj znacznik preferowany dokument handlowy do zamówienia 0 – brak , 1- faktura , 2 – paragon
35	@NumerPrzesylki	Varchar(50)	"	Numer przesyłki
36	@IdOperatoraPrzesylki	Numeric		Identyfikator operatora z tabeli MAG_OPERATOR_PRZESYLKI

Przykład wywołania procedury.

```
EXEC RM_ZatwierdzZamowienie 600,219,12,"<auto>          ","ZO
####/$$",1,1,2,0,1,1,78348,78348,0,2,0,"",1,0,1,0," ",0,1,0,0,0,"",1,"",",",0,"przelew",0,0,"",0
```

5. Zamykanie oraz anulowanie zamówienia

Zamykanie zamówień lub anulowanie polega na zapisaniu w zamówieniu odpowiedniego stanu i realizowane jest procedurą:

5.1 Anulowanie /zamykanie zamówień – opis procedury

Procedura i jej parametry

```
EXEC RM_UstawStanZamowien @id_zamowienia, @id_sesji,
@tryb_inf_realiz , @STATUS_ZAM
```

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_zamowienia,	Numeric		identyfikator z tabeli ZAMOWIENIE nagłówka zamówienia. Jeśli podamy wartość zero (0) to procedura będzie poszukiwała identyfikatorów zamówień w tabeli ZAZNACZONE
2.	@id_sesji	Numeric		identyfikator użytkownika, który zaznaczył zamówienia do zmiany stanu. Używane w przypadku gdy użytkownik zaznaczył zamówienia i @id_zamowienia=0

3.	@tryb_inf_realiz	Tinyint	0	obecnie zawsze równe 0
4.	@status_zam	Char(1)		wartość 'A' jeśli chcemy anulować zamówienia lub 'Y' jeśli zamówienia mają być oznaczone jako zamknięte

5.2 Wycofanie Anulowania/zamknięcia – opis procedury

Procedura i jej parametry

EXEC RM_PrzywrocStanZamowien
 @id_zamowienia,
 @ID_SESJI,
 @id_firmy,
 @STAN_ZAM

Opis parametrów

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_zamowienia,	Numeric		identyfikator z tabeli ZAMOWIENIE nagłówka zamówienia. Jeśli podamy wartość zero (0) to procedura będzie poszukiwała identyfikatorów zamówień w tabeli ZAZNACZONE
2.	@id_sesji	Numeric		identyfikator użytkownika, który zaznaczył zamówienia do zmiany stanu. Używane w przypadku gdy użytkownik zaznaczył zamówienia i @id_zamowienia=0
3.	@id_firmy	Tinyint		identyfikator z tabeli FIRMA, w kontekście której wystawione zostały zamówienia.
4.	@status_zam	Char(1)		wartość 'A' jeśli chcemy anulować zamówienia lub 'Y' jeśli zamówienia mają być oznaczone jako zamknięte

5.3 Przykład dodawania zamówienia

```
if exists (select 1 from sysobjects where name =
'MAGEXT_Generuj_ZD_dla_ZaznaczonychPoz' and type = 'P')
drop procedure MAGEXT_Generuj_ZD_dla_ZaznaczonychPoz
go
create procedure MAGEXT_Generuj_ZD_dla_ZaznaczonychPoz
@id_firmy numeric,
@id_magazynu numeric,
@id_obiektu numeric,
@id_uzytkownika numeric,
@kod_kontekstu int,
@id_obiektu_nadrzednego numeric
```

```

as
declare @errmsg varchar(255) ,
        @id_kontrahenta int,
        @id_zamowienia int,
        @data int,
        @kod_vat varchar(2),
        @id int,
        @id_artykulu int,
        @zo_ilosc numeric (18,4),
        @zo_id_artykulu int ,
        @cena_netto numeric (18,4),
        @cena_brutto numeric (18,4),
        @zo_przelicznik numeric (16,6),
        @zo_jednostka varchar(10),
        @id_typu int,
        @format_num varchar(50),
        @okresnumeracji tinyint ,
        @num_auto tinyint ,
        @num_niezalezny tinyint,
        @max_autonumer int ,
        @id_pracownika int,
        @RetStat T_CHECK_TYP,
        @ID_POZYCJI_ZAMOWIENIA T_ID_TYP,
        @suma_netto numeric (16,4),
        @suma_brutto numeric (16,4),
        @suma_netto_wal numeric (16,4),
        @suma_brutto_wal numeric (16,4),
        @vat_zakupu varchar(3),
        @pole1 varchar(100),
        @pole2 varchar(100),
        @pole4 varchar(100),
        @uwagi_ZD varchar(1000),
        @numer_zd varchar(100),
        @kontr_zo_nazwa varchar(200)

begin
    set xact_abort on
    set transaction isolation level REPEATABLE READ
    begin transaction

    --przykład odczytania kontrahenta
    select @id_kontrahenta = id_kontrahenta
    from dbo.ZAMOWIENIE with(nolock) where ID_ZAMOWIENIA =
@id_objektu_nadrzednego

    --konwersja daty do postaci magowej
    set @data = round(cast(getdate() as real),0,1)+36163

    --dodanie nagłówka zamówienia
    EXEC dbo.RM_DodajZamowienie_Server @id_firmy,
        @id_kontrahenta,
        @id_magazynu, 2 ,
        @data,
        @id_uzytkownika,
        0,
        'Netto',
        1 ,
        @id_zamowienia OUTPUT

    --dodanie pozycji zamówienia
    declare pozycje_zam cursor local fast_forward for

```

```

select z.id, pz.id_artykulu, pz.zamowiono,
       pz.id_artykulu, a.cena_zakupu_netto,
       a.cena_zakupu_brutto, pz.przelicznik,
       pz.jednostka, a.vat_zakupu, pz.pole1 , pz.pole2, pz.POLE4
from dbo.zaznaczone z inner join DBO.pozycja_zamowienia pz
  on z.id=pz.id_pozycji_zamowienia
  inner join artykul a
  on pz.id_artykulu=a.id_artykulu
where z.id_sesji = @id_uzytkownika and z.uzycie = 212

open pozycje_zam
while 1=1
  begin
    FETCH NEXT FROM pozycje_zam INTO
    @id, @id_artykulu, @zo_ilosc,
    @zo_id_artykulu, @cena_netto,
    @cena_brutto, @zo_przelicznik,
    @zo_jednostka, @vat_zakupu , @pole1 , @pole2, @pole4
    if @@fetch_status<>0 break

    EXEC @RetStat = dbo.RM_DodajPozycjeZamowienia_Server
      @ID_POZYCJI_ZAMOWIENIA OUTPUT,
      @id_zamowienia,
      @id_artykulu,
      @vat_zakupu, --@kod_vat,
      @zo_ilosc,
      0, --@zrealizowano,
      0, --@zarezerwowano,
      0, --@do_rezerwacji,
      @cena_netto,
      @cena_brutto,
      0, --@cena_netto_wal,
      0, --@cena_brutto_wal,
      @zo_przelicznik,
      @zo_jednostka,
      0, --@narzut,
      '', --@opis,
      2, --@znak_narzutu,
      0, --@trybrejestracji,
      0, --@id_dostawy_rez,
      0 --@id_wariantu_produkту

    update dbo.POZYCJA_ZAMOWIENIA
    set POLE1 = @pole1, POLE2 = @pole2, POLE4 = @pole4
    where ID_POZYCJI_ZAMOWIENIA = @ID_POZYCJI_ZAMOWIENIA

    IF @RetStat = 0 GOTO error

  end
close pozycje_zam

--odczytanie id_typu dla zamówienia
select @id_typu = ID_TYPU from TYP_DOKUMENTU_MAGAZYNOWEGO where
NAZWA = 'Zamówienie do dostawcy' and ID_FIRMY = @id_firmy

--ustalenie numeracji zamówienia
EXEC dbo.JL_PobierzFormatNumeracji_Server
  @id_firmy ,
  2, --@dokument,

```

```

        @id_typu,
        @id_magazynu , --@id_zasobu ,
        @format_num OUTPUT,
        @okresnumeracji OUTPUT,
        @num_auto OUTPUT,
        @num_niezalezny OUTPUT

    --ustalenie pracownika do dokumentu
    if not exists (select 1 from dbo.PRACOWNIK where ID_UZYTKOWNIKA =
@id_uzytkownika and ID_FIRMY = @id_firmy)
        select @id_pracownika = 0
    else
        select @id_pracownika=id_pracownika from dbo.PRACOWNIK where
ID_UZYTKOWNIKA = @id_uzytkownika and ID_FIRMY = @id_firmy

    --sumowanie wartości zamówienia
    exec dbo.RM_SumujZamowienie_Server @id_zamowienia , @suma_netto
OUTPUT, @suma_brutto OUTPUT, @suma_netto_wal OUTPUT, @suma_brutto_wal
OUTPUT

    --aktualizacja nagłówka zamówienia
    update dbo.ZAMOWIENIE
    set WARTOSC_BRUTTO = @suma_brutto,
        WARTOSC_NETTO = @suma_netto,
        WARTOSC_BRUTTO_WAL = @suma_brutto_wal,
        WARTOSC_NETTO_WAL = @suma_netto_wal
    where ID_ZAMOWIENIA = @id_zamowienia

    --przykładowe uwagi do zamówienia przekazane do zmiennej
    select @uwagi_ZD = 'Zamówienie wygenerowano na podstawie : ' +
NUMER + ' (' +RTRIM(nazwa) + ')'
    from zamowienie z with (nolock) inner join KONTRAHENT k with
(nolock)
    on k.ID_KONTRAHENTA = z.ID_KONTRAHENTA
    where ID_ZAMOWIENIA = @id_obiektu_nadrzednego

    --zatwierdzenie zamówienia
    EXEC dbo.RM_ZatwierdzZamowienie_Server
        @id_zamowienia,
        @id_kontrahenta,
        @id_typu,
        '<auto>',--@numer,
        @format_num,
        @okresnumeracji,
        @num_auto,
        @num_niezalezny,
        @max_autonumer, --@autonumer,
        @id_firmy,
        @id_magazynu,
        @data,
        @data, --@data_realizacji,
        0, --@zaliczka,
        0, --@priorytet,
        2, --@auto_rezerwacja,
        '', --@nr_zam_klienta,
        2, --@typ,
        @id_pracownika,
        1, --@przelicznik_wal,
        0, --@data_kurs_wal,
        '', --@sym_wal,
        0, --@dok_wal,

```

```

1, --@flaga_stanu,
0, --tryb rejestracji
0, --@rabat_narzut,
2, --@znak_rabatu,
@uwagi_ZD --@uwagi,
'',
'',
0,
'przelew',
14

if @@trancount>0 commit transaction
goto Koniec
Error:
raiserror (@errmsg,16,1)
if @@trancount>0 rollback tran
goto Koniec
Koniec:
set transaction isolation level READ COMMITTED
return
end
go

```

6. Dodawanie asortymentu

Dodawanie nowych asortymentów nie ma swojej procedury składowej , należy dodawać je z za pomocą zwykłych zapytań SQL – czyli komendą insert into artykuł (...) values (..) .

Pole id_artykulu nie ma ustawionej właściwości kolumny IDENTITY, więc powinno się odczytać maksymalną wartość ostatniego id przy pomocy np.:

```
SELECT .... @id_a = max(id_artykulu) ... FROM ARTYKUL
```

7. Dodawanie kontrahenta

Dodawanie nowych kontrahentów nie ma swojej procedury składowej , należy dodawać je z za pomocą zwykłych zapytań SQL – czyli komendą insert into kontrahent (...) values (..) .

Pole id_kontrahenta nie ma ustawionej właściwości kolumny IDENTITY, więc powinno się odczytać maksymalną wartość ostatniego id przy pomocy np.:

```
SELECT .... @id_k = max(id_kontrahenta) ... FROM KONTRAHENT
```

8. Procedury pomocnicze dokumentów.

W tym rozdziale zostaną opisane procedury WAPRO Maga przydatne przy zatwierdzaniu dokumentów.

8.1 Funkcja obliczająca wartość brutto dbo.JL_Func_ObliczBrutto()

Funkcja ta pozwala obliczyć np.: cenę brutto na podstawie ceny netto Procedura ta pozwoli na policzenie ceny mechanizmami WAPRO Maga, tak by uniknąć błędów przeliczeń na formatkach

Funkcja i jej parametry

dbo.JL_Func_ObliczBrutto (@cena_netto numeric(16,2) , @stawka_vat char(3))

Stawka VAT powinna znajdować się na liście stawek w tabeli STAWKA_VAT

przykładowe wywołanie:

```
select dbo.JL_Func_ObliczBrutto(12.12,'23')
```

Funkcja przydatna przy dokumentach obliczanych w ewidencji netto

8.2 Funkcja obliczająca wartość netto dbo.JL_Func_ObliczNetto()

Funkcja ta pozwala obliczyć np.: cenę netto na podstawie ceny brutto Procedura ta pozwoli na policzenie ceny mechanizmami WAPRO Maga, tak by uniknąć błędów przeliczeń na formatkach

Funkcja i jej parametry

dbo.JL_Func_ObliczNetto (@cena_brutto numeric(16,2) , @stawka_vat char(3))

Stawka VAT powinna znajdować się na liście stawek w tabeli STAWKA_VAT

przykładowe wywołanie:

```
select dbo.JL_Func_ObliczNetto(12.30,'23')
```

Funkcja przydatna przy dokumentach obliczanych w ewidencji brutto

8.3 Pobieranie parametrów z konfiguracji firmy - dbo.AP_PobierzParaKonf()

Procedura ta pozwala pobrać z konfiguracji użytkownika lub konfiguracji firmy ustawienia.

Procedura i jej parametry

```
exec AP_PobierzParamKonf @typ,@id_firmy,@Kod,@wartosc
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@Typ	Char(1)		Parametr ustalający jaki to jest parametr 'u' – konfiguracja użytkownika 's'- konfiguracja firmy
2.	@IdObiektu	Numeric		W przypadku @typ = 's' identyfikator firmy z której pobieramy parametr w przypadku @typ = 'u' id_uzytkownika z tabeli UZYTKOWNIK
3.	@Kod	Varchar(20)		Wartość kolumny KOD z tabeli POZYCJAKONFIG

4.	@wartosc	Varchar(100) OUTPUT		Parametr OUTPUT zwracający wartość parametru
5.	@Id_firmy	Numeric	0	W przypadku @Typ = 'u' musi być podany identyfikator firmy.

Przykładowe wywołanie do pobrania parametru konfiguracyjnego firmy

```
exec AP_PobierzParamKonf 's',@id_firmy,'BlokDokMagPoZatwFak',@war_par output
```

Przykładowe wywołanie do pobrania parametru konfiguracyjnego użytkownika

```
exec AP_PobierzParamKonf 'u',@id_sesji,'PorOstatCenaDlaKontr',@war_par output,@id_firmy
```

8.4 Dodawanie informacji do logu aktywności dbo.AP_ZapiszDoLogu ()

Procedura ta pozwala zapisać do logu aktywności informacje np.: o dodaniu dokumentu.

Procedura i jej parametry

```
exec AP_ZapiszDoLogu  
@id_firmy,@id_magazynu,@id_uzytkownika,@Id_zapisu,@Typzap,@KodOperacji,@komunikatLog,@sys
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	numeric		identyfikator firmy, w której rejestrowany jest log
2.	@id_magazynu	Numeric		identyfikator magazynu, w którym rejestrowany jest log
3.	@id_uzytkownika	Numeric		Identyfikator użytkownika z tabeli UŻYTKOWNIK
4.	@Id_zapisu	Numeric		Identyfikator zapisu do którego tyczy się log np.: id_dokumentu_handlowego
5.	@Typzap	Varchar(10)		Pole to przyjmuje wartości: ARTYKUL - artykuł DOK_FIN – dok. finansowy DOK_HAN – dok. handlowy DOK_MAG – dok magazynowy DOK_OFE – oferta DOK_PROD – zlecenie produkcyjne DOK_SAD – dokument SAD DOK_ZAM – zamówienie KONTRAHENT - kontrahent MAGAZYN – magazyn

				RACHFIR – rachunek bankowy firmy RACHKONTR – rachunek bankowy kontrahenta ROZRACH - rozrachunek SERWIS – dokumenty zlec. serwisowego
6.	@Kod_oper	Tinyint		1 – dodawanie 2- modyfikacja 3- usuwanie
7.	@identyf_opis	Varchar(100)		Opis w logu
8.	@sys	Tinyint		W przypadku WAPRO Mag to jest cyfra 3

Przykładowe wywołanie zapisu do loga

```
select @komunikatLog = left('Zaimportowano dokument z pliku: '+cast(ltrim(rtrim(@NumerFZ)) as varchar(20)),100)
```

```
exec AP_ZapiszDoLogu  
@id_firmy,@id_magazynu,@id_uzytkownika,@identyfikator_dok,'HAN_CSV',1,@komunikatLog,3
```

8.5 Sumowanie korekty

Procedura i jej parametry

```
EXEC JL_SumujDokumentKorekty  
@id_dok_handl,  
@rodzaj_dok,  
@nr_filtra
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_dok_handl	Int		Identyfikator dokumentu handlowego do sumowania czyli id dokumentu handlowego z tabeli DOKUMENT_HANDLOWY
2.	@rodzaj_dok	Char(2)		Dla korekty zakupu= 'z' dla korekty sprzedaży='s'
3.	@Brutto_netto	Char(6)		W zależności od ewidencji dokumentu przyjmowane są dwie wartości 'Netto' lub 'Brutto'

Zwrotnie otrzymujemy w poleceniu select :

```
select @suma_netto, @suma_brutto
```

Przykładowe wywołanie

```
EXEC JL_SumujDokumentKorekty 1234,'s','Netto'
```

9. Wykorzystanie tabeli zaznaczone.

Odczyt informacji z tabeli

Tabela zaznaczone przechowuje identyfikatory zaznaczonych rekordów na liście. W zależności od użytkownika i listy nabierają znaczenia dwie najważniejsze kolumny ID_SESJI oraz UZYCIE. Podczas każdego otwarcia okna tabela w kontekście użytkownika i listy jest czyszczona. Po zamknięciu okna tabela nie jest usuwana.

Budowa tabeli zaznaczone:

Lp.	Nazwa kolumny	Typ danych	Opis
1.	Id_sesji	Numeric	Identyfikator użytkownika z tabeli UŻYTKOWNIK
2.	Uzycie	Int	Identyfikator listy 1- Dokumenty magazynowe 3- dokumenty finansowe 4- dokumenty handlowe 5 – rozrachunki 6- dokumenty SAD 11-pozycje dokumentu magazynowego 14-pozycje dokumentu handlowego 15- dokumenty intrastat 16- dokumenty akcyzowe 49-zlecenia produkcyjne 50-pozycja zlecenia 102-artykuł 120-kontrahent 202-zamowienia 212-pozycje zamówienia
3	Id	Numeric	Identyfikator zaznaczonego rekordu
4.	ILOSC	Decimal(16,4)	Pole z ilością wpisaną podczas włączonej opcji (grupowe wpisywanie)
5	ID2	Numeric	
6.	ID_AUTO	Numeric	Identyfikator rekordu w tabeli zaznaczone

Jeśli na liście dla użycia brakuje jakiejś informacji lub pojawiają się kolejne listy to sprawdzenie „uzycia” jest bardzo proste – zaznaczamy rekord na liście i sprawdzamy zawartość tabeli, oczywiście dobrze jest wcześniej oczyścić tabele. Proszę takie testy przeprowadzać na bazie testowe lub demonstracyjnej

10. Dodawanie dokumentu zlecenia produkcyjnego

Większość danych i obsługę dokumentu realizują procedury składowane. W przypadku zleceń produkcyjnych wykorzystujemy AP_DodajDokProd_Server, JL_PobierzFormatNumeracji_Server, AP_ZatwierdzDokProd.

10.1 Dodawanie nagłówka zlecenia produkcyjnego

Procedura i jej parametry

```
EXEC [dbo].[AP_DodajDokProd_Server]
  @IdFirmy,
  @IdMagazynu,
  @data,
  @Rodzaj,
  @IdUzytkownika,
  @idzlec OUTPUT
```

Opis parametrów

Lp.	Nazwa kolumny	Typ danych	Wartość domyślna	Opis
1.	@IdFirmy	Numeric		identyfikator firmy, w której rejestrowany jest dokument
2.	@IdMagazynu	Numeric		identyfikator magazynu w kontekście, którego rejestrowany jest dokument
3	@data	Int		data wystawienia dokumentu w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
4.	@rodzaj	Varchar(10)		W przypadku zlecenia montażu wartość 'Montaż' , przy demontażu „
5	@IdUzytkownika	Numeric		Identyfikator użytkownika z tabeli UŻYTKOWNIK
6.	@idzlec	Numeric OUTPUT		Zmienna zwraca identyfikator dodanego nagłówka zlecenia

10.2 Dodawanie pozycji zlecenia produkcyjnego

W przypadku zleceń produkcyjnych pozycje dodawane są zwykłym insertem do bazy danych.

```
INSERT INTO dbo.POZYCJE_ZLECENIA (.....)
```

```
SELECT (.....)
```

Tabela pozycja zamówienia nie ma identyfikatora pozycji id_pozycji automatycznie nadawanego należy więc go samemu wyliczyć np.: funkcją max()

```
select @id_pozycji = max(ID_POZYCJI)+1 from POZYCJE_ZLECENIA
```

10.3 Procedura JL_PobierzFormatNumeracji_Server

Procedura i jej parametry

```
EXEC JL_PobierzFormatNumeracji_Server  
@id_firmy ,  
@dokument,  
@id_typu,  
@id_zasobu ,  
@format_num OUTPUT,  
@okresnumeracji OUTPUT,  
@parametr1 OUTPUT,  
@parametr2 OUTPUT
```

Opis parametrów wejściowych

Lp.	Nazwa parametru	Typ danych	Wartość domyślna	Wartości parametru
1.	@id_firmy	Int		Id_firmy z tabeli firma
2.	@dokument	Tinyint	7	
3.	@id_typu	Int		Id_typu z tabeli typ_dokumentu_magazynowego
4.	@id_zasobu	Int		id_magazynu z tabeli magazyn

Procedura ta zwraca 4-ry parametry wyjściowe (OUTPUT), które należy przekazać do procedury AP_ZatwierdzDokProd jako :

Opis parametrów wyjściowych

Lp.	Nazwa parametru	Typ zwracanej zmiennej	Podstawić jako parametr w procedurze JL_zatwierdzDokumentMagazynowy
1.	@format_numeracji	Varchar(50)	@num_format
2.	@okresnumeracji	Tinyint	@num_okresNumeracji
3.	@Parametr1	Tinyint	@num_auto
4.	@Parametr2	TinyInt	@num_niezalezny

10.4 Zatwierdzanie zlecenia produkcyjnego

Procedura i jej parametry

```
EXEC dbo.AP_ZatwierdzDokProd
    @Idzlec,
    @Idtypu,
    @Numer,
    @num_format,
    @okresnumeracji,
    @num_auto,
    @num_niezalezna,
    @idfirmy,
    @IdMagazynu,
    @Data,
    @Rodzaj,
    @Uwagi,
    @Flaga,
    @IdUzytkownika,
    @IdPracownika,
    @FIFO_LIFO,
    @przeliczaj_ceny,
    @Data_rw,
    @Data_pw,
    @Data_od,
    @Data_do,
    @Czas_od,
    @Czas_do,
    @Id_kontrahenta,
    @Id_zamowienia,
```

Opis parametrów

Lp.	Nazwa kolumny	Typ danych	Wartość domyślna	Opis
1.	@Idzlec	Numeric		wewnętrzne id dokumentu
2.	@Idtypu	Numeric		- identyfikator typu dokumentu (pole id_typu z tabeli TYP_DOKUMENTU_MAGAZYNOWEGO)
3.	@Numer	Varchar(30)		numer dokumentu handlowego. Wartość = '<auto>' w trybie automatycznej numeracji,
4.	@num_format	Varchar(30)		maska formatu numeracji
5.	@okresnumeracji	Tinyint		ciągły ,miesięczny ,roczny, dzienny (1,0,2,3)
6.	@num_auto	Tinyint		numeracja automatyczna = 1 lub ręczna 2

7.	@num_niezalezna	tinyint		numeracja zależna od magazynu lub nie(1-zależna , 2 -niezależna)
8.	@IdFirmy	Numeric		identyfikator firmy, w której rejestrowany jest dokument
9.	@IdMagazynu	Numeric		identyfikator magazynu w kontekście, którego rejestrowany jest dokument
10	@data	Int		data wystawienia dokumentu w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
11.	@rodzaj	Varchar(10)		W przypadku zlecenia montażu wartość 'Montaż' , przy demontażu „
12.	@uwagi	Varchar(500)		Uwagi do zlecenia produkcyjnego
13.	@Flaga	Tinyint		Flaga stanu - w przypadku dodawania zlecenia ustawiamy na 1 - dodawanie
14.	@IdUzytkownika	Numeric		Identyfikator użytkownika z tabeli UŻYTKOWNIK
15.	@IdPracownika	Numeric		pole z tabeli PRACOWNIK – identyfikator pracownika wystawiającego / odpowiedzialnego za dokument
16.	@FIFO_LIFO	Varchar(10)		Jeśli pole uste to nie wygenerują się RW+PW w przypadku uzupełnienia wartością FIFO oba dokumenty powstaną
17.	@przeliczaj_ceny	tinyint		0 Nie przeliczaj 1 Przeliczaj ceny
18.	@Data_rw	Int		data wystawienia dokumentu RWw formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
19.	@Data_pw	Int		data wystawienia dokumentu PW w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
20.	@Data_od	Int	0	Czas trwania -data w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
21.	@Data_do	Int	0	Czas trwania -data w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
22.	@Czas_od	Int	0	Czas trwania -time w formacie wewnętrznym
23.	@Czas_do	Int	0	Czas trwania -time w formacie wewnętrznym
24.	@Id_kontrahenta	Numeric	0	pole z tabeli KONTRAHENT – kontrahent powiązany z dokumentem
25.	@Id_zamowienia	Numeric	0	pole z tabeli ZAMOWIENIE – zamówienie powiązane z dokumentem

10.5 Przykładowe wywołanie procedur do założenia zlecenia produkcyjnego montażu.

```
if exists (select 1 from sysobjects where name = 'MAGSRC_GenZLM' and type = 'P')
    drop procedure MAGSRC_GenZLM
go
create procedure MAGSRC_GenZLM
@IdFirmy numeric, @IdMagazynu numeric, @IdObjektu numeric, @IdObjektuNadrzednego numeric,
@IdUzytkownika numeric, @Wynik1 varchar(255) OUTPUT
as
declare @errmsg varchar(255)
begin
    set xact_abort on
    set transaction isolation level REPEATABLE READ
    begin transaction

--deklaracja zmiennych
declare @data int ,
        @idzlec numeric,
        @IdArtykulu numeric ,
        @IdWariantu numeric,
        @jednostka varchar(10),
        @przelicznik numeric(16,6),
        @id_pozycji numeric ,
        @Ilosc_zlecona numeric(16,4),
        @id_Typu_zlec int ,
        @Id_Pracownika numeric ,
        @num_format varchar(50),
        @okresnumeracji tinyint,
        @num_auto tinyint,      -- podstawiamy pod @num_auto
        @num_niezalezna tinyint,
        @NumerRW varchar(50)

--ustalenie daty dokumentu na podstawie daty systemowej
select @data = round(cast(getdate() as real),0,1)+36163

--ustalenie typu zlecenia
select @id_Typu_zlec = id_typu from typ_dokumentu_magazynowego t with (nolock)
where sygnatura = 'ZLM' and id_firmy = @Idfirmy

--odszukanie pracownika
if exists ( SELECT 1 from dbo.Pracownik p with (nolock) where id_uzytkownika =
@IdUzytkownika)
    begin
select @Id_Pracownika = id_pracownika from dbo.Pracownik p with (nolock) where
id_uzytkownika = @IdUzytkownika
    end
else
    select @Id_Pracownika = 0

--dodajemy nagłówek zlecenia produkcyjnego
EXEC [dbo].[AP_DodajDokProd_Server]
@IdFirmy,@IdMagazynu,@data,'Montaż',@IdUzytkownika, @idzlec OUTPUT

--odszukanie asortymentu po indeksie
select
        @IdArtykulu = a.id_artykulu,
        @IdWariantu = pw.id_wariantu, --,
        @jednostka = pw.jednostka_prod,
        @przelicznik = pw.przelicznik_prod
```



```

from dbo.artykul a with (nolock) inner join dbo.produkt_wariant pw
      on a.id_artykulu = pw.id_artykulu
where id_magazynu = @Idmagazynu and a.ID_ARTYKULU = @IdArtykulu
      and pw.domyslny = 1

```

```

--obliczenie maksymalnego id
if exists (select top 1 1 from POZYCJE_ZLECENIA)
select @id_pozycji = max(ID_POZYCJI)+1 from POZYCJE_ZLECENIA
else
select @id_pozycji = 1

```

```

INSERT INTO dbo.POZYCJE_ZLECENIA
      (ID_POZYCJI,
      ID_ZLECENIA,
      ID_ARTYKULU,
      ID_WARIANTU,
      JEDNOSTKA,
      PRZELICZNIK,
      STATUS,
      ILOSC,
      DO_PRODUKCJI,
      WYPRODUKOWANO,
      OPIS,
      NR_SERII,
      DATA_WAZNOSCI,
      ID_POZYCJI_MONTAZU,

```

```

      POLE1,POLE2,POLE3,POLE4,POLE5,POLE6,POLE7,POLE8,POLE9,
      POLE10)

```

```

select
      @id_pozycji,
      @idzlec,
      @IdArtykulu,
      @IdWariantu,
      @jednostka,
      @przelicznik,
      'R',
      @Ilosc_zlecona,
      @Ilosc_zlecona,
      0,
      '--opis
      ', --nr_serii
      0, --data_waznosci
      0, --id_pozycji_montazu
      ..... --pola dodatkowe

```

----dodanie pozycji do zaznaczonych na potrzeby zatwierdzania zlecenia jeśli generujemy zlecenie bez RW/PW nie jest to wymagane

```

insert into dbo.zaznaczone (id_sesji, uzycie, id, ilosc, id2)
select @IdUzytkownika, 50, @id_pozycji, NULL,0

```

--ustalenie numeracji

```

EXEC dbo.JL_PobierzFormatNumeracji_Server @idfirmy,
      7, --zlecenie produkcyjne
      @id_Typu_zlec,
      @Idmagazynu,
      @num_format OUTPUT,
      @okresnumeracji OUTPUT,
      @num_auto OUTPUT,
      @num_niezalezna OUTPUT

```

```

--zatwierdzenie dokumentu zlecenia produkcyjnego
EXEC dbo.AP_ZatwierdzDokProd
    @idzlec,
    @id_Typu_zlec,
    '<auto>',
    @num_format,
    @okresnumeracji,
    @num_auto,
    @num_niezalezna,
    @idfirmy,
    @IDMAGAZYNU,
    @DATA,
    'Montaż',
    ", --uwagi
    1, --flaga stanu
    @IdUzytkownika,
    @Id_pracownika,
    ", --'FIFO', --", --jak nie ma fifo - tylko pusto to nie powstają dokumenty RW/PW
    1,--przeliczaj ceny
    @DATA, --data rw
    @DATA, --data pw
    0, --data od
    0,--data do
    0, --czas od
    0, --czas do
    0,--id_kontrahenta
    0 --id_zamowieia

    if @errmsg <> " goto error

    declare @Uwagi varchar(1000),
            @komunikat varchar(100)

    update ZLECENIE_PRODUKCYJNE set
        UWAGI = 'Uwagi... '
    where ID_ZLECENIA = @idzlec

---dobrze jest zapisać dodanie takie zlecenia w logu aktywności aplikacji
select @komunikat = left('Zlecenie: ' + numer + ' zaimportowane',100)
from dbo.zlecenie_produkcyjne with (nolock) where id_zlecenia = @idzlec

exec AP_ZapiszDoLogu
    @idfirmy,@idmagazynu,@iduzytkownika,@idzlec,'DOK_PROD',1,@komunikat,3

```

```

if @@trancount>0 commit transaction
goto Koniec
Error:
raiserror (@errmsg,16,1)
if @@trancount>0 rollback tran
goto Koniec
Koniec:
set transaction isolation level READ COMMITTED
return
end
go

```

11. Dodawanie dokumentu oferty.

Większość danych i obsługę dokumentu realizują procedury składowane. W przypadku ofert wykorzystujemy RM_OFERTA_Dodaj_Server, RM_OFERTA_DodajPozycje_Server, RM_OFERTA_Zatwierdz_Server.

11.1 Procedura dodawania nagłówka oferty

Opis parametrów procedury RM_OFERTA_Dodaj_Server

Lp.	Nazwa kolumny	Typ danych	Wartość domyślna	Opis
1.	@Id_Firmy	Numeric		identyfikator firmy, w której rejestrowany jest dokument oferty
2.	@Id_Kontrahenta	Numeric		Identyfikator kontrahenta z tabeli KONTRAHENT
3.	@Id_magazynu	Numeric		identyfikator magazynu w kontekście, którego rejestrowany jest dokument oferty
4.	@id_typu	Numeric		- identyfikator typu oferty (pole id_typu z tabeli TYP_DOKUMENTU_OFERTY
5.	@Data	Int		data wystawienia oferty w formacie wewnętrznym = round(convert(real,getdate()),0) + 36163
6.	@Semafor	Numeric		identyfikator użytkownika z tabeli UŻYTKOWNIK
7.	@flaga_stanu	Tinyint		1 – dodawanie oferty
8.	@id_zapisu_wersji	Numeric		
9.	@trybrejestracji	Tinyint	0	Pole wewnętrzne
10.	@Brutto_netto			Obliczanie oferty wg - 'Netto' lub 'Brutto'
11.	@Id_oferty	Numeric OUTPUT		Parametr OUTPUT, zwraca id_oferty – id nagłówka w bazie danych – id_oferty z tabeli OFERTA

11.2 Procedura dodawania pozycji oferty

Opis parametrów procedury RM_OFERTA_DodajPozycje_Server

Lp.	Nazwa kolumny	Typ danych	Wartość domyślna	Opis
1.	@ID_pozycji_oferty	Numeric OUTPUT		Parametr OUTPUT, zwraca identyfikator dodawanej pozycji oferty – id_pozycji_oferty z tabeli

				POZYCJA_OFERTY
2.	@Id_oferty	Numeric		- identyfikator nagłówka oferty do której dodawana jest pozycja
3.	@id_firmy	Numeric		identyfikator firmy (z tabeli FIRMA), w której dodawana jest pozycja oferty
4.	@Id_magazynu	Numeric		identyfikator magazynu (z tabeli MAGAZYN), w którym powstaje pozycja oferty
5.	@Id_artykułu	Numeric		identyfikator artykułu z tabeli ARTYKUL , którego dotyczy pozycja
6.	@Kod_vat	Varchar(3)		kod stawki VAT pozycji
7.	@cena_netto	Decimal(14,4)		Cena netto pozycji na ofercie
8.	@cena_brutto	Decimal(14,4)		Cena brutto pozycji na ofercie
9.	@cena_netto_wal	Decimal(14,4)		Cena netto walutowa pozycji na ofercie
10.	@cena_brutto_wal	Decimal(14,4)		Cena brutto walutowa pozycji na ofercie
11.	@przelicznik	Decimal(16,6)		przelicznik jednostki towaru z tabeli jednostka względem jednostki magazynowania,
12.	@jednostka	Varchar(10)		Skrót jednostki z tabeli jednostka
13.	@opis	Varchar(500)		Opis pozycji oferty
14.	@rabat	Decimal(8,4)		Wartość narzutu /rabatu
15.	@znak_rabatu	Tinyint		znak (+) = 0 lub rabatu (-) = 2
16.	@trybrejestracji	Tinyint		0
17.	@id_wariantu_produkту	Numeric	0	Wskazanie na identyfikator wariantu produktu dla asortymentu. (aktualnie nie żywane)
18.	@id_zapisu_pierwszej	Numeric	0	Identyfikator pierwszej oferty.
19	@znacznik_ceny	Char(1)		Znacznik ceny dla pozycji jako litera: k kartotekowa p promocyjna i indywidualna g dla grupy cenowej m modyfikowana ręcznie

11.3 Procedura zatwierdzająca dokument oferty

Opis parametrów procedury RM_OFERTA_Zatwierdz_Server

Lp.	Nazwa kolumny	Typ danych	Wartość domyślna	Opis
1.	@Id_oferty	Numeric		identyfikator nagłówka oferty która jest zatwierdzana
2.	@id_firmy	Numeric		identyfikator firmy, w której rejestrowany jest dokument oferty
3.	@id_magazynu	Numeric		identyfikator magazynu w kontekście, którego rejestrowany jest dokument oferty
4.	@id_kontrahenta	Numeric		identyfikator kontrahenta (z tabeli KONTRAHENT), którego dotyczy oferta
5.	@id_typu	Numerci		- identyfikator typu oferty (pole id_typu z tabeli TYP_DOKUMENTU_OFERTY
6.	@obliczanie_wg_cen	Char(6)		Obliczanie oferty wg - 'Netto' lub 'Brutto'
7.	@rodzaj_oferty	Char(1)	S	Aktualnie tylko oferty sprzedaży -znacznik 's'
8.	@numer	Varchar(30)		Wartość <auto> - automatyczne nadanie numeru oferty
9.	@nazwa	Varchar(300)		Nazwa oferty
10	@data	Int		data wystawienia oferty w formacie wewnętrznym = round(convert(real.getdate()),0) + 36163
11.	@data_waznosci_od	Int		„data od” ważności oferty w formacie wewnętrznym = round(convert(real.getdate()),0) + 36163
12.	@data_waznosci_do	Int		„data do” ważności oferty w formacie wewnętrznym = round(convert(real.getdate()),0) + 36163
13.	@plan_termin_realizacji	Int		Planowany termin ważności oferty w formacie wewnętrznym = round(convert(real.getdate()),0) + 36163
14.	@przelicznik_wal	Decimal(14,6)		wartość przelicznika waluty w przypadku zamówień walutowych, przy złotówkowych zamówieniach = 1
15.	@data_kurs_wal	Int		Data kursu waluty formacie wewnętrznym = round(convert(real.getdate()),0) + 36163
16.	@Sym_wal	Char(3)		symbol waluty, puste dla PLN,
17.	@dok_wal	Tinyint		= 1 dokument walutowy,=0 dokument złotówkowy (w walucie bazowej)
18.	@id_pracownika	Numeric		pole z tabeli PRACOWNIK – identyfikator pracownika wystawiającego / odpowiedzialnego za obsługę zamówienia

19.	@Id_uzytkownika	Numeric		id_uzytkownika zatwierdzającego dokument oferty z tabeli UZYTKOWNIK
20.	@trybrejestracji	Tinyint	0	
21.	@uwagi	Varchar(1000)		uwagi tekstowe do złożonej oferty
22.	@errmsg	Varchar(255) OUTPUT		Zmiana zwróci błąd z procedury
23.	@OsobaZamawiajaca	Varchar(101)		Imię i nazwisko w tabeli kontakt
24.	@id_kontaktu	Numeric		Identyfikator kontaktu z tabeli KONTAKT
25.	@FormaPlatnosci	Varchar(50)		Nazwa formy płatności z tabeli FORMY_PLATNOSCI
26.	@DniPlatnosci	Int		Ilość dni dla formy płatności
27.	@FakturaParagon	Tinyint	0	
28.	@Id_powodu_odrzucenia	numeric	0	

11.4 Przykładowe wywołanie procedur zakładających dokument oferty.

```

if exists (select 1 from sysobjects where name = 'MAGSRC_DodajOferte' and type = 'P')
    drop procedure MAGSRC_DodajOferte
go
create procedure MAGSRC_DodajOferte
@IdFirmy numeric, @IdMagazynu numeric, @IdObjektu numeric, @IdUzytkownika numeric,
@Wynik1 varchar(255) OUTPUT
as
declare @errmsg varchar(255)
begin
set xact_abort on
set transaction isolation level REPEATABLE READ
begin transaction

    declare @IdOferty numeric
    declare @IdKontrahenta numeric
    declare @IdTypuOF int
    declare @Data int
    declare @RetStat int
    declare @brutto_netto varchar(10)
    declare @flaga_stanu int
    declare @id_zapisu_wersji int
    declare @trybrejestracji int
    declare @IdPozycjiOferty numeric
    declare @IdArtykulu numeric
    declare @KodVat varchar(5)

```

```

declare @CenaNetto numeric(16,2)
declare @CenaBrutto numeric(16,2)
declare @Jednostka varchar(10)
declare @RodzajOferty char(1)
declare @DataTerminRealizacji int

--ustalenie kontrahenta do oferty
select @IdKontrahenta = ID_KONTRAHENTA
      from kontrahent where NAZWA = 'TEST' and ID_FIRMY = @IdFirmy

--ustalenie typu dla oferty
select @IdTypuOF = ID_TYPU
      from TYP_DOKUMENTU_OFERTY with (nolock) where id_firmy=@IdFirmy and
substring(Rodzaj,1,1)='R'

--ustalenie danych dodatkowych
set @data = round(cast(GETDATE() as real),0,1)+36163
set @DataTerminRealizacji = @Data + 15

set @brutto_netto = 'Netto'
set @flaga_stanu = 1
set @id_zapisu_wersji = 0
set @trybrejestracji = 0
set @RodzajOferty = 's'

--dodanie nagłówka oferty.
exec @RetStat = dbo.RM_OFERTA_Dodaj_Server
      @idfirmy           ,
      @idkontrahenta     ,
      @idmagazynu        ,
      @IdTypuOF          ,
      @data               ,
      @IdUzytkownika     ,
      @flaga_stanu       ,
      @id_zapisu_wersji ,
      @trybrejestracji   ,
      @brutto_netto      ,
      @IdOferty OUTPUT

if @RetStat = 0
begin
select @errmsg = 'Błąd dodawania nagłówka oferty'
goto ERROR
end

--dodatkowy update na rekordzie oferty
update OFERTA set
      ID_KONTRAHENTA = @IdKontrahenta,
      RODZAJ_OFERTY = 's',
      FORMA_PLATNOSCI = 'przelew',
      pole1='',pole2='',pole3='',pole4='',pole5='',pole6='',
      pole7='',pole8='',pole9='',pole10=''
where id_pozycji_oferty=@IdPozycjiOferty

where ID_OFERTY = @IdOferty

--odszukanie asortymentu
select @IdArtykulu = a.IDARTYKULU,
      @KodVat = a.VATSprzedazy,
      @CenaNetto = a.CenaNettoSprzedazyDomylna,
      @CenaBrutto = a.CenaBruttoSprzedazyDomylna,
      @Jednostka = j.SKROT

```

```

from dbo.WIDOK_ARTYKUL a inner join dbo.JEDNOSTKA j
    on a.IdJednostkiPodstawowej = j.ID_JEDNOSTKI
where NrKatalogowy = 'Tow000001' and IdMagazynu = @IdMagazynu

```

```

--dodanie pozycji na ofertę
EXEC [dbo].[RM_OFERTA_DodajPozycje_Server]
    @IdPozycjiOferty OUTPUT,
    @IdOferty,
    @IdFirmy,
    @IdMagazynu,
    @IdArtykulu,
    @KodVat,
    1, --ilość
    @CenaNetto,
    @CenaBrutto,
    0, --cena netto wal
    0, --cena brutto wal
    1, --przelicznik
    @Jednostka,
    '', --opis
    0, --rabat
    2, -- znak rabatu
    0, --trybrejestracji
    0, --id_wariantu_produkta
    0, --id_zapisu pierwszej
    'm' --znacznik ceny

```

```

--ustalenie pól dodatkowych dla pozycji oferty
update Pozycja_oferty
set pole1='',pole2='',pole3='',pole4='',pole5='',pole6='',
pole7='',pole8='',pole9='',pole10=''
where id_pozycji_oferty=@IdPozycjiOferty

```

```

--zatwierdzenie oferty

```

```

exec [dbo].[RM_OFERTA_Zatwierdz_Server]
    @IdOferty,
    @IdFirmy,
    @IdMagazynu,
    @IdKontrahenta,
    @IdTypuOF,
    @brutto_netto,
    @RodzajOferty,
    '<auto>', --numer
    '', --nazwa
    @Data, --data oferty
    @Data, --oferta data waznosci od
    @Data, --oferta data waznosci do
    @DataTerminRealizacji,
    1, --przelicznik walutowy
    0, --data kurs wal
    '', --symbol wal
    0, --czy dokument wal
    0, --id_pracownika
    @IdUzytkownika,
    0, --trybrejestracji
    '', --uwagi
    @errmsg OUTPUT,
    '', --osoba zamawiająca
    0, --idkontraktu
    'przelew', --forma płatności
    0, --dni płatności

```



```

0, --faktura paragon
0 --id powodu odrzucenia

if @@trancount>0 commit transaction
goto Koniec
Error:
raiserror (@errmsg,16,1)
if @@trancount>0 rollback tran
goto Koniec
Koniec:
set transaction isolation level READ COMMITTED
return
end
go

```

12. Zmiany w dokumencie

12.1 Poprawki w dokumentacji (2012-10-25)

- a) Dostosowanie do wersji 7.80.0

12.2 Poprawki w dokumentacji (2015-07-01)

- a) Dostosowanie do wersji 8.00.8
 - aktualizacja procedur zatwierdzających dokument faktury oraz generowania rozliczenia
 - aktualizacja procedur zatwierdzających dokument faktury korekty oraz generowania rozliczenia
 - aktualizacja procedury zatwierdzającej dokument magazynowy
 - aktualizacja procedury zatwierdzającej dokument zamówienia

12.3 Poprawki w dokumentacji (2016-01-20)

- a) Dostosowanie do wersji 8.10.4
 - aktualizacja procedury zatwierdzającej pozycje zamówienia
 - aktualizacja procedury zatwierdzającej dokument handlowy
 - aktualizacja procedury dodającej dokument korekty

12.4 Poprawki w dokumentacji (2017-03-10)

- a) Dostosowanie do wersji 8.21
 - aktualizacja procedury zatwierdzającej zamówienia

12.5 Poprawki w dokumentacji (2018-10-10)

- a) Dostosowanie do wersji 8.40
 - aktualizacja procedury zatwierdzającej pozycje dokumentu magazynowego i handlowego

- aktualizacja procedury dodającej nagłówek dokumentu handlowego - korekta.
- aktualizacja procedury zatwierdzającej pozycję korekty
- aktualizacja procedury zatwierdzającej dokument finansowy
- aktualizacja procedury zatwierdzającej pozycje zamówienia
- opis procedur dodających zlecenie produkcyjne
- modyfikacja przykładu skryptu SQL dodającego dokument FZUE.
- opis procedur/funkcji dodatkowych

12.6 Poprawki w dokumentacji (2018-10-25)

- dodanie opisu procedur dodających dokument ofert

12.7 Poprawki w dokumentacji (2020-03-06)

a) Dostosowanie do wersji 8.50

- aktualizacja proceduruy zatwierdzającej dokument magazynowy o obsługę MPP.
- aktualizacja proceduruy zatwierdzającej dokument handlowy o obsługę MPP.

12.8 Drobne poprawki w dokumentacji (2020-12-04)