

```
use TPP;
go

-- PRIVILEGE AND USER MANAGEMENT
--
create procedure list_users_xml
as
with XMLNAMESPACES (DEFAULT 'http://ruch.bytom.pjwstk.edu.pl/TPPDB/AuthorizationService')
    select Login "@Login", Imie "@FirstName", Nazwisko "@LastName"
    from Uzytkownik
    for XML PATH('UserDetails'), root ('UserList')
go

--
create procedure get_user ( @user_login varchar(30) )
as
    select Login, Imie, Nazwisko, Email
    from Uzytkownik
    where Login = @user_login
go

--
create procedure validate_password(@login varchar(30), @pass varchar(25), @res bit OUTPUT)
as
begin
declare @c int = 0;
    select @c = COUNT(*) from Uzytkownik where Login = @login and Haslo = HashBytes('SHA1',@pass) and Status < 0;
if (@c = 1) set @res = 1; else set @res = 0;
end;
go

--
-- Error codes:
-- 1 login already in use
-- 2 email already in use
-- 3 obligatory parameter empty (length 0)
create procedure create_user_account(@user_login varchar(30), @user_password varchar(20), @user_email      <
    varchar(50), @user_first_name varchar(30), @user_last_name varchar(50), @activation_code varchar(10), @  <
    result int OUTPUT)
as
begin

    declare @email_title as varchar (120);
    declare @email_body as varchar (500);
    set @result = 0;

    if ( LEN(@user_login)=0 or LEN(@user_password)=0 or LEN(@user_email) = 0 )
    begin
        set @result = 3;
        return;
    end;

    if exists(select * from Uzytkownik where Login = @user_login)
    begin
        set @result = 1;
        return;
    end;
    if exists(select * from Uzytkownik where Email = @user_email)
    begin
        set @result = 2;
        return;
    end;

    insert into Uzytkownik ( Login, Haslo, Email, Imie, Nazwisko, Kod_aktywacji ) values ( @user_login,      <
    HashBytes('SHA1',@user_password), @user_email, @user_first_name, @user_last_name, @activation_code );

    set @email_title = 'TPP Database account activation for ' + @user_login;
    set @email_body = 'Your account with login '+@user_login+' has been created successfully.'+CHAR(13)
    +' To activate your account use the following link: ...AccountActivation.aspx?login='+@user_login+'&code <
end;
```

```

        ='+@activation_code+CHAR(13)
        +'Alternatively, use activation code for login '+@user_login +' and activation code ' + @activation_code
        +' to perform the activation manually '
        +CHAR(13)+'on the webpage https://v21.pjwstk.edu.pl/HMDB/UserAccountCreation.aspx or through your client
        application.';

        exec msdb.dbo.sp_send_dbmail @profile_name='HMDB_Mail',
        @recipients=@user_email,
        @subject= @email_title,
        @body= @email_body;
        return 0;
end;
go

--
-- Error codes:
-- 1 authentication negative
create procedure activate_user_account(@user_login varchar(30), @activation_code varchar(10), @result int
    OUTPUT)
as
begin
    set @result = 0;

    if not exists(select * from Uzytkownik where Login = @user_login and Kod_aktywacji = @activation_code )
        begin
            set @result = 1;
            return;
        end;
    update Uzytkownik set Status = 1 where Login = @user_login and Kod_aktywacji = @activation_code;

    return 0;
end;
go

--
-- Error codes:
-- 1 email not found
-- 2 account not yet activated
-- 3 obligatory parameter empty (length 0)
-- 4 HMDB propagate requested but account of this email missing
create procedure forgot_password(@user_email varchar(50), @activation_code varchar(20), @result int OUTPUT)
as
begin

    declare @email_title as varchar (120);
    declare @email_body as varchar (500);
    declare @link_command as varchar(30);
    declare @user_login as varchar(30);

    set @link_command = '';
    set @result = 0;

    if ( LEN(@user_email)=0 )
        begin
            set @result = 3;
            return;
        end;

    if not exists(select * from Uzytkownik where Email = @user_email)
        begin
            set @result = 1;
            return;
        end;
    if exists(select * from Uzytkownik where Email = @user_email and Status = 0)
        begin
            set @result = 2;
            return;
        end;

    select @user_login = Login from Uzytkownik where Email = @user_email;

    update Uzytkownik set Kod_aktywacji = @activation_code, Status = 3 where Email = @user_email;

```

```

set @email_title = 'TPP Database password reset request confirmation';
set @email_body = 'The password reset keycode for your account with login '+@user_login+' has been
generated.'+CHAR(13)
+'To set the new password authenticate using the login '+@user_login+' and the temporary password ' + @
activation_code+' using '
+CHAR(13)+'the webpage https://???/UserAccountUpdate.aspx .';

exec msdb.dbo.sp_send_dbmail @profile_name='HMDB_Mail',
@recipients=@user_email,
@subject= @email_title,
@body= @email_body;
return 0;
end;
go

--
-- Error codes:
-- 1 authentication failed
-- 2 email already in use
-- 3 obligatory parameter empty (length 0)
create procedure update_user_account(@user_login varchar(30), @user_password varchar(20), @
user_new_password varchar(20), @user_email varchar(50), @user_first_name varchar(30), @user_last_name
varchar(50), @result int OUTPUT)
as
begin

    set @result = 0;

    if ( LEN(@user_login)=0 or LEN(@user_password)=0 )
    begin
        set @result = 3;
        return;
    end;
    if not exists(select * from Uzytkownik where Login = @user_login and Haslo = HashBytes('SHA1',@
user_password) and Status > 0 )
    begin
        if exists(select * from Uzytkownik where Login= @user_login and Kod_Aktywacji = @user_password
and Status = 3 )
        begin
            update Uzytkownik set Haslo = HashBytes('SHA1',@user_new_password), Status = 1 where
Login = @user_login;
            return;
        end
        else
        begin
            set @result = 1;
            return;
        end;
    end;
    if exists(select * from Uzytkownik where Login != @user_login and Email = @user_email)
    begin
        set @result = 2;
        return;
    end;

    if ( @user_first_name != '-nochange-' )
    begin
        update Uzytkownik
        set Email = @user_email, Imie = @user_first_name, Nazwisko = @user_last_name where Login = @
user_login;
    end;
    if ( @user_new_password != '-nochange-' )
    begin
        update Uzytkownik set Haslo = HashBytes('SHA1',@user_new_password) where Login = @user_login;
    end;

    return @result;
end;
go

--
create procedure reset_password(@login varchar(30), @old varchar(25), @new varchar(25), @res int OUTPUT )

```

```

as
begin
if not exists(select * from Uzytkownik where Login = @login and Haslo = HashBytes('SHA1',@old))
begin
    set @res = 1;
    return;
end;
update Uzytkownik set Haslo = HashBytes('SHA1',@new) where Login = @login and Haslo = HashBytes('SHA1',@old) ✓
;
return 0;
end;
go

create function identify_user( @user_login varchar(50) )
returns int
as
begin
    declare @user_id int;
    set @user_id = 0;
    select @user_id = IdUzytkownik from Uzytkownik where Login = @user_login;
    return @user_id;
end;
go

-- SLOWNIKOWANIE (listowanie opcji i walidacje)
-- =====

create function validate_input_int( @table_name varchar(30), @attr_name varchar(50), @value tinyint )
returns bit
as
begin
if @value IS NULL return 1;
return ( select count(*) from SlownikInt where Tabela = @table_name and Atrybut = @attr_name and Klucz = @value ); ✓
end
go

create procedure get_enumeration_int( @table_name varchar(30), @attr_name varchar(50) )
as
    select Klucz as Value, Definicja as Label from SlownikInt where Tabela = @table_name and Atrybut = @attr_name ✓
go

create function validate_input_decimal( @table_name varchar(30), @attr_name varchar(50), @value decimal(3,1) )
returns bit
as
begin
if @value IS NULL return 1;
return ( select count(*) from SlownikDecimal where Tabela = @table_name and Atrybut = @attr_name and Klucz = @value ); ✓
end
go

create procedure get_enumeration_decimal( @table_name varchar(30), @attr_name varchar(50) )
as
    select Klucz as Value, Definicja as Label from SlownikDecimal where Tabela = @table_name and Atrybut = @attr_name ✓
go

create function validate_input_varchar( @table_name varchar(30), @attr_name varchar(50), @value varchar(20) )
returns bit
as
begin
if @value IS NULL return 1;
return ( select count(*) from SlownikVarchar where Tabela = @table_name and Atrybut = @attr_name and Klucz = @value ); ✓
end
go

create procedure get_enumeration_varchar( @table_name varchar(30), @attr_name varchar(50) )

```

```

as
    select Klucz as Value, Definicja as Label from SlownikVarchar where Tabela = @table_name and Atrybut =
    @attr_name
go

-- WGRYWANIE DANYCH I WALIDACJA
-- =====

--
-- @result codes: 0 = OK, 1 = patient exists while update existing not allowed, 2 = validation failed - see
message
create procedure update_patient ( @NumerPacjenta varchar(20), @NazwaGrupy varchar(3), @RokUrodzenia
smallint, @MiesiacUrodzenia tinyint, @Plec tinyint, @Lokalizacja varchar(10), @LiczbaElektrod tinyint, @
allow_update_existing bit, @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin
    declare @update bit;
    set @result = 0;
    set @update = 0;
    set @message = '';
    if(( select count(*) from Pacjent where NumerPacjenta = @NumerPacjenta )>0 )
    begin
        if ( @allow_update_existing = 0 )
        begin
            set @result = 1;
            return;
        end
        else set @update = 1;
    end;

    if( dbo.validate_input_varchar('Pacjent', 'NazwaGrupy', @NazwaGrupy) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'Nazwa grupy - invalid enum value.';
        return;
    end;

    if( dbo.validate_ext_bit( @Plec) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'Plec - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if( dbo.validate_year_month( @RokUrodzenia, @MiesiacUrodzenia) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'MiesiacUrodzenia / RokUrodzenia - invalid value.';
        return;
    end;

    if( dbo.validate_input_varchar('Pacjent', 'Lokalizacja', @Lokalizacja) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'Lokalizacja - invalid enum value.';
        return;
    end;

    if(@update = 0)
        insert into Pacjent (NumerPacjenta, NazwaGrupy, RokUrodzenia, MiesiacUrodzenia, Plec, Lokalizacja,
        LiczbaElektrod )
        values (@NumerPacjenta, @NazwaGrupy, @RokUrodzenia, @MiesiacUrodzenia, @Plec, @
        Lokalizacja, @LiczbaElektrod );
    else
        update Pacjent
        set RokUrodzenia = @RokUrodzenia, MiesiacUrodzenia = @MiesiacUrodzenia, Plec = @Plec, Lokalizacja =
        @Lokalizacja,
        LiczbaElektrod = @LiczbaElektrod
        where NumerPacjenta = @NumerPacjenta;
end;
go

```

```
drop procedure update_patient_1
--
-- @result codes: 0 = OK, 1 = patient exists while update existing not allowed, 2 = validation failed - see ✓
-- message, 3 = login user not found
create procedure update_patient_1 (    @NumerPacjenta varchar(20), @NazwaGrupy varchar(3), @RokUrodzenia ✓
    smallint, @MiesiacUrodzenia tinyint, @Plec tinyint, @Lokalizacja varchar(10), @LiczbaElektrod tinyint, @ ✓
    allow_update_existing bit, @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin
    declare @update bit;
    set @result = 0;
    set @update = 0;
    declare @userid int;
    set @userid = 0;
    set @message = '';
    if( ( select count(*) from Pacjent where NumerPacjenta = @NumerPacjenta )>0 )
    begin
        if ( @allow_update_existing = 0 )
        begin
            set @result = 1;
            return;
        end
        else set @update = 1;
    end;

    if( dbo.validate_input_varchar('Pacjent', 'NazwaGrupy', @NazwaGrupy) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'Nazwa grupy - invalid enum value.';
        return;
    end;

    select @userid = dbo.identify_user(@actor_login);

    if ( @userid = 0 )
    begin
        set @result = 3;
        set @message = 'Unknown login ' + @actor_login ;
        return;
    end;

    if( dbo.validate_ext_bit( @Plec) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'Plec - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if( dbo.validate_year_month( @RokUrodzenia, @MiesiacUrodzenia) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'MiesiacUrodzenia / RokUrodzenia - invalid value.';
        return;
    end;

    if( dbo.validate_input_varchar('Pacjent', 'Lokalizacja', @Lokalizacja) = 0 )
    begin
        set @result = 2;
        set @message = @message + 'Lokalizacja - invalid enum value.';
        return;
    end;

    if(@update = 0)
        insert into Pacjent (NumerPacjenta, NazwaGrupy, RokUrodzenia, MiesiacUrodzenia, Plec, Lokalizacja, ✓
        LiczbaElektrod, OstatniaZmiana, Wprowadzil, Zmodyfikowal )
        values (@NumerPacjenta, @NazwaGrupy, @RokUrodzenia, @MiesiacUrodzenia, @Plec, @ ✓
        Lokalizacja, @LiczbaElektrod, getdate(), @userid, @userid    );
    else
        update Pacjent
        set RokUrodzenia = @RokUrodzenia, MiesiacUrodzenia = @MiesiacUrodzenia, Plec = @Plec, Lokalizacja = ✓
        @Lokalizacja,
        LiczbaElektrod = @LiczbaElektrod,
        Zmodyfikowal = @userid,
```

```

        OstatniaZmiana = getdate()
        where NumerPacjenta = @NumerPacjenta;
end;
go

create function validate_ext_bit ( @val tinyint )
returns bit
as
begin
return case @val
        when 0 then 1
        when 1 then 1
        when 2 then 1
        else 0 end;
end;
go

create function validate_year_month ( @year smallint, @month tinyint )
returns bit
as
begin
        declare @res bit;
        set @res = 1;
        if( @year < 1920 or @year > 2050 or @month < 1 or @month > 12) set @res = 0;
        return @res;
end;
go

--
-- @result codes: 0 = OK, 1 = visit already exists while run in no-update mode, exist 2 = validation failed ✓
-- see message, 3 = patient of this number not found, 4 = user login unknown
create procedure update_examination_questionnaire_partA (@NumerPacjenta varchar(20), @RodzajWizyty decimal ✓
(2,1),
@DataPrzyjecia date, @DataOperacji date, @DataWypisu date, @MasaCiala decimal(4,1),
@Wyksztalcenie tinyint, @Rodzinnosc tinyint, @RokZachorowania smallint, @MiesiacZachorowania tinyint, ✓
@RokBadania smallint, @MiesiacBadania tinyint,
@PierwszyObjaw tinyint,
@Drzenie tinyint,
@Sztynosc tinyint,
@Spowolnienie tinyint,
@ObjawyInne tinyint,
@ObjawyInneJakie varchar(80),
@CzasOdPoczObjDoWlLDopy tinyint, @DyskinezyObecnie tinyint, @DyskinezyOdLat decimal(3,1), @ ✓
FluktuacjeObecnie tinyint, @FluktuacjeOdLat decimal(3,1),
@CzasDyskinez decimal(3,1), @CzasOFF decimal(3,1), @PoprawaPoLDopie tinyint,
@allow_update_existing bit,
@actor_login varchar(50),
@result int OUTPUT, @visit_id int OUTPUT, @message varchar(200) OUTPUT )
as
begin
        declare @patient_id int;
        declare @user_id int;
        declare @update int;
        set @patient_id = 0;
        set @result = 0;
        set @message = '';
        set @update = 0;
        select @patient_id = IdPacjent from Pacjent where NumerPacjenta = @NumerPacjenta;

        if(@patient_id = 0)
        begin
                set @result = 3;
                set @message = 'patient of this number not found';
                return;
        end;
        select @user_id = dbo.identify_user(@actor_login);
        if(@user_id = 0)
        begin
                set @result = 4;
                set @message = 'user of this login not found';
                return;
        end;
end;

```

```
if (select count(*) from Wizyta where IdPacjent = @patient_id and RodzajWizyty = @RodzajWizyty ) > 0
begin
    if (@allow_update_existing = 0)
        begin
            set @result = 1;
            return;
        end;
    set @update = 1;
    select @visit_id = IdWizyta from Wizyta where IdPacjent = @patient_id and RodzajWizyty = @
    RodzajWizyty;
end;

-- validations

if dbo.validate_input_decimal('Wizyta', 'RodzajWizyty', @RodzajWizyty) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute RodzajWizyty';
    return;
end;
if dbo.validate_input_int('Wizyta', 'Wyksztalcenie', @Wyksztalcenie) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute Wyksztalcenie';
    return;
end;
if( dbo.validate_ext_bit( @Rodzinnosc) = 0 )
begin
    set @result = 2;
    set @message = 'Rodzinnosc - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_year_month( @RokZachorowania, @MiesiacZachorowania) = 0 )
begin
    set @result = 2;
    set @message = 'RokZachorowania / MiesiacZachorowania - invalid value.';
    return;
end;

if( dbo.validate_year_month( @RokBadania, @MiesiacBadania) = 0 )
begin
    set @result = 2;
    set @message = 'RokBadania / MiesiacBadania - invalid value.';
    return;
end;

if dbo.validate_input_int('Wizyta', 'PierwszyObjaw', @PierwszyObjaw) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute PierwszyObjaw';
    return;
end;

if( dbo.validate_ext_bit( @Drzenie) = 0 )
begin
    set @result = 2;
    set @message = 'Drzenie - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @Sztynosc) = 0 )
begin
    set @result = 2;
    set @message = 'Sztynosc - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @Spowolnienie) = 0 )
begin
    set @result = 2;
    set @message = 'Spowolnienie - invalid value. Allowed: 0, 1, 2.';
    return;
end;
```



```

end;

if( dbo.validate_ext_bit( @ObjawyInne) = 0 )
begin
    set @result = 2;
    set @message = 'ObjawyInne - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @DyskinezyObecnie) = 0 )
begin
    set @result = 2;
    set @message = 'DyskinezyObecnie - invalid value. Allowed: 0, 1, 2.';
    return;
end;
if( dbo.validate_ext_bit( @FluktuacjeObecnie) = 0 )
begin
    set @result = 2;
    set @message = @message + 'FluktuacjeObecnie - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @PoprawaPoLDopie) = 0 )
begin
    set @result = 2;
    set @message = 'PoprawaPoLDopie - invalid value. Allowed: 0, 1, 2.';
    return;
end;

-- /validations
-- depending on update
if(@update = 0)
begin
    insert into Wizyta (      RodzajWizyty, IdPacjent, DataPrzyjecia, DataOperacji, DataWypisu, MasaCiala,
    Wyksztalcenie, Rodzinnosc, RokZachorowania, MiesiacZachorowania, RokBadania, MiesiacBadania,
    PierwszyObjaw,
    Drzenie, Sztywnosc, Spowolnienie, ObjawyInne, ObjawyInneJakie,
    CzasOdPoczObjDoWlLDopy, DyskinezyObecnie,
    DyskinezyOdLat, FluktuacjeObecnie, FluktuacjeOdLat, CzasDyskinez, CzasOFF,
    PoprawaPoLDopie,
    Wprowadzil, Zmodyfikowal, OstatniaZmiana )
    values (      @RodzajWizyty, @patient_id, @DataPrzyjecia, @DataOperacji, @DataWypisu, @
    MasaCiala, @Wyksztalcenie, @Rodzinnosc, @RokZachorowania, @MiesiacZachorowania, @RokBadania, @
    MiesiacBadania, @PierwszyObjaw,
    @Drzenie, @Sztywnosc, @Spowolnienie, @ObjawyInne, @ObjawyInneJakie,
    @CzasOdPoczObjDoWlLDopy, @DyskinezyObecnie,
    @DyskinezyOdLat, @FluktuacjeObecnie, @FluktuacjeOdLat, @CzasDyskinez, @
    CzasOFF, @PoprawaPoLDopie,
    @user_id, @user_id, getdate() ) set @visit_id = SCOPE_IDENTITY();
end;
else
begin
    update Wizyta
    set DataPrzyjecia = @DataPrzyjecia, DataOperacji = @DataOperacji, DataWypisu = @DataWypisu,
    MasaCiala = @MasaCiala, Wyksztalcenie = @Wyksztalcenie, Rodzinnosc = @Rodzinnosc, RokZachorowania = @
    RokZachorowania,
    MiesiacZachorowania = @MiesiacZachorowania, RokBadania = @RokBadania, MiesiacBadania = @
    MiesiacBadania, PierwszyObjaw = @PierwszyObjaw,
    Drzenie = @Drzenie,
    Sztywnosc = @Sztywnosc,
    Spowolnienie = @Spowolnienie,
    ObjawyInne = @ObjawyInne,
    ObjawyInneJakie = @ObjawyInneJakie,
    CzasOdPoczObjDoWlLDopy = @CzasOdPoczObjDoWlLDopy, DyskinezyObecnie = @DyskinezyObecnie,
    DyskinezyOdLat = @DyskinezyOdLat, FluktuacjeObecnie = @FluktuacjeObecnie,
    FluktuacjeOdLat = @FluktuacjeOdLat,
    CzasDyskinez = @CzasDyskinez, CzasOFF = @CzasOFF,
    PoprawaPoLDopie = @PoprawaPoLDopie,
    Zmodyfikowal = @user_id, OstatniaZmiana = getdate()
    where RodzajWizyty = @RodzajWizyty and IdPacjent = @patient_id;
end;
return;
end;

```

```
go

-- drop procedure update_examination_questionnaire_partB
--
-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 = user login unknown
create procedure update_examination_questionnaire_partB (@IdWizyta int,
  @PrzebyteLeczenieOperacyjnePD tinyint,
  @Papierosy tinyint,
  @Kawa tinyint,
  @ZielonaHerbata tinyint,
  @Alkohol tinyint,
  @ZabiegowWZnieczOgPrzedRozpoznaniemPD tinyint,
  @Zamieszkanie tinyint,
  @Uwagi varchar(50),
  @Nadcisnienie tinyint,
  @BlokeryKanWapn tinyint,
  @DominujacyObjawObecnie tinyint,
  @DominujacyObjawUwagi varchar(50),
  @RLS tinyint,
  @ObjawyPsychotyczne tinyint,
  @Depresja tinyint,
  @Otepienie decimal(2,1),
  @Dyzartria tinyint,
  @DysfagiaObjaw tinyint,
  @RBD tinyint,
  @ZaburzenieRuchomosciGalekOcznych tinyint,
  @Apraksja tinyint,
  @TestKlaskania tinyint,
  @ZaburzeniaWechowe tinyint,
  @actor_login varchar(50),
  @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

  declare @user_id int;
  set @user_id = 0;

  set @result = 0;

  select @user_id = dbo.identify_user(@actor_login);
  if(@user_id = 0)
  begin
    set @result = 4;
    set @message = 'user of this login not found';
    return;
  end;

  if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
  begin
    set @result = 3;
    set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar)+' not found';
    return;
  end;

  -- validations

  if dbo.validate_input_int('Wizyta', 'PrzebyteLeczenieOperacyjnePD', @PrzebyteLeczenieOperacyjnePD) = 0
  begin
    set @result = 2;
    set @message = 'Invalid value for attribute PrzebyteLeczenieOperacyjnePD - see enumeration for allowed values';
    return;
  end;

  if dbo.validate_input_int('Wizyta', 'Papierosy', @Papierosy) = 0
  begin
```

```
        set @result = 2;
        set @message = 'Invalid value for attribute Papierosy';
        return;
    end;
    if dbo.validate_input_int('Wizyta', 'Kawa', @Kawa) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute Kawa';
        return;
    end;

    if( dbo.validate_ext_bit( @ZielonaHerbata) = 0 )
    begin
        set @result = 2;
        set @message = 'ZielonaHerbata - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'Alkohol', @Alkohol) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute Alkohol';
        return;
    end;
    if dbo.validate_input_int('Wizyta', 'Zamieszkanie', @Zamieszkanie) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute Zamieszkanie';
        return;
    end;

    if( dbo.validate_ext_bit( @Nadcisnienie ) = 0 )
    begin
        set @result = 2;
        set @message = 'Nadcisnienie - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if( dbo.validate_ext_bit( @BlokeryKanWapn ) = 0 )
    begin
        set @result = 2;
        set @message = 'BlokeryKanWapn - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'DominujacyObjawObecnie', @DominujacyObjawObecnie) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute DominujacyObjawObecnie - see enumeration for allowed values';
        return;
    end;

    if( dbo.validate_ext_bit( @RLS) = 0 )
    begin
        set @result = 2;
        set @message = 'RLS - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if( dbo.validate_ext_bit( @ObjawyPsychotyczne) = 0 )
    begin
        set @result = 2;
        set @message = 'ObjawyPsychotyczne - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if( dbo.validate_ext_bit( @Depresja) = 0 )
    begin
        set @result = 2;
        set @message = 'Depresja - invalid value. Allowed: 0, 1, 2.';
        return;
    end;
```

```

if dbo.validate_input_decimal('Wizyta', 'Otepienie', @Otepienie) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute Otepienie';
    return;
end;

if( dbo.validate_ext_bit( @Dyzartria ) = 0 )
begin
    set @result = 2;
    set @message = 'Dyzartria - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @DysfagiaObjaw ) = 0 )
begin
    set @result = 2;
    set @message = 'Dysfagia objaw - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @RBD ) = 0 )
begin
    set @result = 2;
    set @message = 'RBD - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @ZaburzenieRuchomosciGalekOcznych ) = 0 )
begin
    set @result = 2;
    set @message = 'ZaburzenieRuchomosciGalekOcznych - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @Apraksja ) = 0 )
begin
    set @result = 2;
    set @message = 'Apraksja - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @TestKlaskania ) = 0 )
begin
    set @result = 2;
    set @message = 'TestKlaskania - invalid value. Allowed: 0, 1, 2.';
    return;
end;

if( dbo.validate_ext_bit( @ZaburzeniaWechowe ) = 0 )
begin
    set @result = 2;
    set @message = 'ZaburzeniaWechowe - invalid value. Allowed: 0, 1, 2.';
    return;
end;

update Wizyta
set
    PrzebyteLeczenieOperacyjnePD = @PrzebyteLeczenieOperacyjnePD,
    Papierosy = @Papierosy, Kawa = @Kawa,
    ZielonaHerbata = @ZielonaHerbata,
    Alkohol = @Alkohol,
    ZabiegowWZnieczOgPrzedRozpoznaniemPD = @ZabiegowWZnieczOgPrzedRozpoznaniemPD,
    Zamieszkanie = @Zamieszkanie,
    Uwagi = @Uwagi,
    Nadcisnienie = @Nadcisnienie,
    BlokeryKanWapn = @BlokeryKanWapn,
    DominujacyObjawObecnie = @DominujacyObjawObecnie,
    DominujacyObjawUwagi = @DominujacyObjawUwagi,

    RLS = @RLS,
    ObjawyPsychotyczne = @ObjawyPsychotyczne,

```

```

        Depresja      = @Depresja,
        Otepienie     = @Otepienie,
        Dyzartria     = @Dyzartria,
        DysfagiaObjaw = @DysfagiaObjaw,
        RBD           = @RBD,
        ZaburzenieRuchomosciGalekOcznych = @ZaburzenieRuchomosciGalekOcznych,
        Apraksja      = @Apraksja,
        TestKlaskania = @TestKlaskania,
        ZaburzeniaWechowe = @ZaburzeniaWechowe,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdWizyta = @IdWizyta;

    return;
end;
go

/*

*/

-- modified: 2013-09-08
-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 = user login unknown
create procedure update_examination_questionnaire_partC (
    @IdWizyta int,
    @Ldopa tinyint,
    @LDopaObecnie smallint,
    @Agonista bit,
    @AgonistaObecnie smallint,
    @Amantadyna bit,
    @AmantadynaObecnie smallint,
    @MAOBinh bit,
    @MAOBinhObecnie smallint,
    @COMTinh bit,
    @COMTinhObecnie smallint,
    @Cholinolityk bit,
    @CholinolitykObecnie smallint,
    @LekiInne bit,
    @LekiInneJakie varchar(50),
    @L_STIMOpis varchar(30),
    @L_STIMAmplitude decimal(5,1),
    @L_STIMDuration decimal(5,1),
    @L_STIMFrequency decimal(5,1),
    @R_STIMOpis varchar(30),
    @R_STIMAmplitude decimal(5,1),
    @R_STIMDuration decimal(5,1),
    @R_STIMFrequency decimal(5,1),
    @Wypis_Ldopa tinyint,
    @Wypis_LDopaObecnie smallint,
    @Wypis_Agonista bit,
    @Wypis_AgonistaObecnie smallint,
    @Wypis_Amantadyna bit,
    @Wypis_AmantadynaObecnie smallint,
    @Wypis_MAOBinh bit,
    @Wypis_MAOBinhObecnie smallint,
    @Wypis_COMTinh bit,
    @Wypis_COMTinhObecnie smallint,
    @Wypis_Cholinolityk bit,
    @Wypis_CholinolitykObecnie smallint,
    @Wypis_LekiInne bit,
    @Wypis_LekiInneJakie varchar(50),
    @Wypis_L_STIMOpis varchar(30),
    @Wypis_L_STIMAmplitude decimal(5,1),
    @Wypis_L_STIMDuration decimal(5,1),
    @Wypis_L_STIMFrequency decimal(5,1),
    @Wypis_R_STIMOpis varchar(30),
    @Wypis_R_STIMAmplitude decimal(5,1),
    @Wypis_R_STIMDuration decimal(5,1),
    @Wypis_R_STIMFrequency decimal(5,1),

```

```

    @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
    begin
        set @result = 3;
        set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar) + ' not found';
        return;
    end;

    update Wizyta
    set
        Ldopa = @Ldopa,
        LDopaObecnie = @LDopaObecnie,
        Agonista = @Agonista,
        AgonistaObecnie = @AgonistaObecnie,
        Amantadyna = @Amantadyna,
        AmantadynaObecnie = @AmantadynaObecnie,
        MAOBinh = @MAOBinh,
        MAOBinhObecnie = @MAOBinhObecnie,
        COMTinh = @COMTinh,
        COMTinhObecnie = @COMTinhObecnie,
        Cholinolityk = @Cholinolityk,
        CholinolitykObecnie = @CholinolitykObecnie,
        LekiiInne = @LekiiInne,
        LekiiInneJakie = @LekiiInneJakie,
        L_STIMOpis = @L_STIMOpis,
        L_STIMAmplitude = @L_STIMAmplitude,
        L_STIMDuration = @L_STIMDuration,
        L_STIMFrequency = @L_STIMFrequency,
        R_STIMOpis = @R_STIMOpis,
        R_STIMAmplitude = @R_STIMAmplitude,
        R_STIMDuration = @R_STIMDuration,
        R_STIMFrequency = @R_STIMFrequency,
        Wypis_Ldopa = @Wypis_Ldopa,
        Wypis_LDopaObecnie = @Wypis_LDopaObecnie,
        Wypis_Agonista = @Wypis_Agonista,
        Wypis_AgonistaObecnie = @Wypis_AgonistaObecnie,
        Wypis_Amantadyna = @Wypis_Amantadyna,
        Wypis_AmantadynaObecnie = @Wypis_AmantadynaObecnie,
        Wypis_MAOBinh = @Wypis_MAOBinh,
        Wypis_MAOBinhObecnie = @Wypis_MAOBinhObecnie,
        Wypis_COMTinh = @Wypis_COMTinh,
        Wypis_COMTinhObecnie = @Wypis_COMTinhObecnie,
        Wypis_Cholinolityk = @Wypis_Cholinolityk,
        Wypis_CholinolitykObecnie = @Wypis_CholinolitykObecnie,
        Wypis_LekiiInne = @Wypis_LekiiInne,
        Wypis_LekiiInneJakie = @Wypis_LekiiInneJakie,
        Wypis_L_STIMOpis = @Wypis_L_STIMOpis,
        Wypis_L_STIMAmplitude = @Wypis_L_STIMAmplitude,
        Wypis_L_STIMDuration = @Wypis_L_STIMDuration,
        Wypis_L_STIMFrequency = @Wypis_L_STIMFrequency,
        Wypis_R_STIMOpis = @Wypis_R_STIMOpis,
        Wypis_R_STIMAmplitude = @Wypis_R_STIMAmplitude,
        Wypis_R_STIMDuration = @Wypis_R_STIMDuration,
        Wypis_R_STIMFrequency = @Wypis_R_STIMFrequency,
        Zmodyfikowal = @user_id,

```

```
OstatniaZmiana = getdate()
where IdWizyta = @IdWizyta;

return;
end;
go

-- DEPRECATED. 2013-08-15 (attributes moved to B and H)
-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 = user login unknown
create procedure update_examination_questionnaire_partD (
    @IdWizyta int,
    @PrzebyteLeczenieOperacyjnePD tinyint,
    @Nadcisnienie tinyint,
    @BlokeryKanWapn tinyint,
    @DominujacyObjawObecnie tinyint,
    @DominujacyObjawUwagi varchar(50),
    @BadanieWechu bit,
    @WynikWechu tinyint,
    @LimitDysfagii tinyint,
    @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
    begin
        set @result = 3;
        set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar)+' not found';
        return;
    end;

    -- attribute validators

    if dbo.validate_input_int('Wizyta', 'PrzebyteLeczenieOperacyjnePD', @PrzebyteLeczenieOperacyjnePD) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute PrzebyteLeczenieOperacyjnePD - see enumeration for allowed values';
        return;
    end;

    if( dbo.validate_ext_bit( @Nadcisnienie ) = 0 )
    begin
        set @result = 2;
        set @message = 'Nadcisnienie - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if( dbo.validate_ext_bit( @BlokeryKanWapn ) = 0 )
    begin
        set @result = 2;
        set @message = 'BlokeryKanWapn - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'DominujacyObjawObecnie', @DominujacyObjawObecnie) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute DominujacyObjawObecnie - see enumeration for allowed values';
    end;
```

```

        return;
    end;

    if( dbo.validate_ext_bit( @BadanieWechu ) = 0 )
    begin
        set @result = 2;
        set @message = 'BadanieWechu - invalid value. Allowed: 0, 1, 2.';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'WynikWechu', @WynikWechu) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute WynikWechu - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'LimitDysfagii', @LimitDysfagii) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute LimitDysfagii - see enumeration for allowed values';
        return;
    end;

    update Wizyta
    set
        PrzebyteLeczenieOperacyjnePD = @PrzebyteLeczenieOperacyjnePD,
        Nadcisnienie = @Nadcisnienie,
        BlokeryKanWapn = @BlokeryKanWapn,
        DominujacyObjawObecnie = @DominujacyObjawObecnie,
        DominujacyObjawUwagi = @DominujacyObjawUwagi,
        BadanieWechu = @BadanieWechu,
        WynikWechu = @WynikWechu,
        LimitDysfagii = @LimitDysfagii,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdWizyta = @IdWizyta;

    return;
end;
go

-- updated (temp. name) 2013-11-21
-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 =
-- user login unknown
create procedure update_examination_questionnaire_partE_x01 (
    @IdWizyta int,
    @WydzielanieSliny tinyint,
    @Dysfagia tinyint,
    @DysfagiaCzestotliwosc tinyint,
    @Nudnosci tinyint,
    @Zaparcia tinyint,
    @TrudnosciWOddawaniuMoczu tinyint,
    @PotrzebaNaglegoOddaniaMoczu tinyint,
    @NiekompletneOproznieniePecherza tinyint,
    @SlabyStrumienMoczu tinyint,
    @CzestotliwoscOddawaniaMoczu tinyint,
    @Nykturia tinyint,
    @NiekontrolowaneOddawanieMoczu tinyint,
    @Omdlenia tinyint,
    @ZaburzeniaRytmuSerca tinyint,
    @ProbaPionizacyjna tinyint,
    @WzrostPodtlivostiTwarzKark tinyint,
    @WzrostPotliivostiRamionaDlonie tinyint,
    @WzrostPotliivostiBrzuchPlecy tinyint,
    @WzrostPotliivostiKonczynyDolneStopy tinyint,
    @SpadekPodtlivostiTwarzKark tinyint,
    @SpadekPotliivostiRamionaDlonie tinyint,
    @SpadekPotliivostiBrzuchPlecy tinyint,
    @SpadekPotliivostiKonczynyDolneStopy tinyint,
    @NietolerancjaWysokichTemp tinyint,
    @NietolerancjaNiskichTemp tinyint,
    @Lojotok tinyint,

```



```
@SpadekLibido tinyint,
@KlopotyOsiagnieciaErekcji tinyint,
@KlopotyUtrzymaniaErekcji tinyint,
@actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
    begin
        set @result = 3;
        set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar)+' not found';
        return;
    end;

    -- attribute validators

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @WydzielanieSliny) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute WydzielanieSliny - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @Dysfagia) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @Dysfagia - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @DysfagiaCzestotliwosc) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @DysfagiaCzestotliwosc - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @Nudnosci) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute Nudnosci - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @Zaparcia) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute Zaparcia - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @TrudnosciWOddawaniuMoczu) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute TrudnosciWOddawaniuMoczu - see enumeration for allowed values';
        return;
    end;
```

```
if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @PotrzebaNaglegoOddaniaMoczu) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @PotrzebaNaglegoOddaniaMoczu - see enumeration for
allowed values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @NiekompletneOproznieniePecherza) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @NiekompletneOproznieniePecherza - see enumeration for
allowed values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @SlabyStrumienMoczu) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @SlabyStrumienMoczu - see enumeration for allowed values
';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @CzestotliwoscOddawaniaMoczu) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @CzestotliwoscOddawaniaMoczu - see enumeration for
allowed values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @Nykturia) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @Nykturia - see enumeration for allowed values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @NiekontrolowaneOddawanieMoczu) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @NiekontrolowaneOddawanieMoczu - see enumeration for
allowed values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @Omdlenia) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @Omdlenia - see enumeration for allowed values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @ZaburzeniaRytmuSerca) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @ZaburzeniaRytmuSerca - see enumeration for allowed
values';
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @ProbaPionizacyjna) = 0
begin
    set @result = 2;
    set @message = 'Invalid value for attribute @ProbaPionizacyjna - see enumeration for allowed values'
;
    return;
end;

if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @WzrostPodtliwosciTwarzKark) = 0
begin
```

```
        set @result = 2;
        set @message = 'Invalid value for attribute @WzrostPodtliwosciTwarzKark - see enumeration for
allowed values';
        return;
    end;

--
    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @WzrostPotliwosciRamionaDlonie) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @WzrostPotliwosciRamionaDlonie - see enumeration for
allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @WzrostPotliwosciBrzuchPlecy) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @WzrostPotliwosciBrzuchPlecy - see enumeration for
allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @WzrostPotliwosciKonczynyDolneStopy) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @WzrostPotliwosciKonczynyDolneStopy - see enumeration
for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @SpadekPodtliwosciTwarzKark) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @SpadekPodtliwosciTwarzKark - see enumeration for
allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @SpadekPotliwosciRamionaDlonie) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @SpadekPotliwosciRamionaDlonie - see enumeration for
allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @SpadekPotliwosciBrzuchPlecy) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @SpadekPotliwosciBrzuchPlecy - see enumeration for
allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @SpadekPotliwosciKonczynyDolneStopy) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute SpadekPotliwosciKonczynyDolneStopy - see enumeration for
allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @NietolerancjaWysokichTemp) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @NietolerancjaWysokichTemp - see enumeration for allowed
values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @NietolerancjaNiskichTemp) = 0
    begin
        set @result = 2;
```

```

        set @message = 'Invalid value for attribute @NietolerancjaNiskichTemp - see enumeration for allowed
values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @Lojotok) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @Lojotok - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @SpadekLibido) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @SpadekLibido - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @KlopotyOsiagnieciaErekcji) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @KlopotyOsiagnieciaErekcji - see enumeration for allowed
values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', '_ObjawAutonomiczny', @KlopotyUtrzymaniaErekcji) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute @KlopotyUtrzymaniaErekcji - see enumeration for allowed
values';
        return;
    end;

    update Wizyta
    set
        WydzielanieSliny = @WydzielanieSliny,
        Dysfagia = @Dysfagia,
        DysfagiaCzestotliwosc = @DysfagiaCzestotliwosc,
        Nudnosci = @Nudnosci,
        Zaparcia = @Zaparcia,
        TrudnosciWOddawaniuMoczu = @TrudnosciWOddawaniuMoczu,
        PotrzebaNaglegoOddaniaMoczu = @PotrzebaNaglegoOddaniaMoczu,
        NiekompletneOproznieniePecherza = @NiekompletneOproznieniePecherza,
        SlabyStrumienMoczu = @SlabyStrumienMoczu,
        CzestotliwoscOddawaniaMoczu = @CzestotliwoscOddawaniaMoczu,
        Nykturia = @Nykturia,
        NiekontrolowaneOddawanieMoczu = @NiekontrolowaneOddawanieMoczu,
        Omdlenia = @Omdlenia,
        ZaburzeniaRytmuSerca = @ZaburzeniaRytmuSerca,
        ProbaPionizacyjna = @ProbaPionizacyjna,
        WzrostPodtlivostiTwarzKark = @WzrostPodtlivostiTwarzKark,
        WzrostPotlivostiRamionaDlonie = @WzrostPotlivostiRamionaDlonie,
        WzrostPotlivostiBrzuchPlecy = @WzrostPotlivostiBrzuchPlecy,
        WzrostPotlivostiKonczynyDolneStopy = @WzrostPotlivostiKonczynyDolneStopy,
        SpadekPodtlivostiTwarzKark = @SpadekPodtlivostiTwarzKark,
        SpadekPotlivostiRamionaDlonie = @SpadekPotlivostiRamionaDlonie,
        SpadekPotlivostiBrzuchPlecy = @SpadekPotlivostiBrzuchPlecy,
        SpadekPotlivostiKonczynyDolneStopy = @SpadekPotlivostiKonczynyDolneStopy,
        NietolerancjaWysokichTemp = @NietolerancjaWysokichTemp,
        NietolerancjaNiskichTemp = @NietolerancjaNiskichTemp,
        Lojotok = @Lojotok,
        SpadekLibido = @SpadekLibido,
        KlopotyOsiagnieciaErekcji = @KlopotyOsiagnieciaErekcji,
        KlopotyUtrzymaniaErekcji = @KlopotyUtrzymaniaErekcji,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdWizyta = @IdWizyta;

    return;
end;
go

```

```
-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 = user login unknown
create procedure update_examination_questionnaire_partF (
    @IdWizyta int,
    @PDQ39 tinyint,
    @AIMS tinyint,
    @Epworth tinyint,
    @CGI tinyint,
    @FSS decimal(3,1),
    @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
    begin
        set @result = 3;
        set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar)+' not found';
        return;
    end;

    -- attribute validators

    update Wizyta
    set
        PDQ39 = @PDQ39,
        AIMS = @AIMS,
        Epworth = @Epworth,
        CGI = @CGI,
        FSS = @FSS
    where IdWizyta = @IdWizyta;

    return;
end;
go

-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 = user login unknown
create procedure update_examination_questionnaire_partG (
    @IdWizyta int,
    @TestZegara bit,
    @MMSE tinyint,
    @WAIS_R_Wiadomosci tinyint,
    @WAIS_R_PowtarzanieCyfr tinyint,
    @SkalaDepresjiBecka tinyint,
    @TestFluencjiZwierzeta tinyint,
    @TestFluencjiOstre tinyint,
    @TestFluencjiK tinyint,
    @TestLaczeniaPunktowA tinyint,
    @TestLaczeniaPunktowB tinyint,
    @TestUczeniaSlownoSluchowego tinyint,
    @TestStroopa tinyint,
    @TestMinnesota tinyint,
    @InnePsychologiczne varchar(150),
    @OpisBadania varchar(2000),
    @Wnioski varchar(2000),
    @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin
```

```

declare @user_id int;
set @user_id = 0;

set @result = 0;

select @user_id = dbo.identify_user(@actor_login);
if(@user_id = 0)
begin
    set @result = 4;
    set @message = 'user of this login not found';
    return;
end;

if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
begin
    set @result = 3;
    set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar)+' not found';
    return;
end;

update Wizyta
set
    TestZegara = @TestZegara,
    MMSE = @MMSE,
    WAIS_R_Wiadomosci = @WAIS_R_Wiadomosci,
    WAIS_R_PowtarzanieCyfr = @WAIS_R_PowtarzanieCyfr,
    SkalaDepresjiBecka = @SkalaDepresjiBecka,
    TestFluencjiZwierzeta = @TestFluencjiZwierzeta,
    TestFluencjiOstre = @TestFluencjiOstre,
    TestFluencjiK = @TestFluencjiK,
    TestLaczeniaPunktowA = @TestLaczeniaPunktowA,
    TestLaczeniaPunktowB = @TestLaczeniaPunktowB,
    TestUczeniaSlownoSluchowego = @TestUczeniaSlownoSluchowego,
    TestStroopa = @TestStroopa,
    TestMinnesota = @TestMinnesota,
    InnePsychologiczne = @InnePsychologiczne,
    OpisBadania = @OpisBadania,
    Wnioski = @Wnioski,
    Zmodyfikowal = @user_id,
    OstatniaZmiana = getdate()
where IdWizyta = @IdWizyta;

return;
end;
go

-- updated 2013-10-14
-- @result codes: 0 = OK, 3 = visit of this ID not found, exist 2 = validation failed - see message, 4 =
-- user login unknown
create procedure update_examination_questionnaire_parth (
    @IdWizyta int,
    @Holter bit,
    @BadanieWechu bit,
    @WynikWechu tinyint,
    @LimitDysfagii tinyint,
    @pH_metriaPrzełyku bit,
    @SPECT bit,
    @MRI bit,
    @MRIwynik varchar(2000),
    @USGszrodmozgowia bit,
    @USGwynik tinyint,
    @Genetyka bit,
    @GenetykaWynik varchar(50),
    @Surowica bit,
    @SurowicaPozostało varchar(50),
    @Ferrytyna decimal(7,3),
    @CRP decimal(7,3),
    @NTproCNP decimal(7,3),

```

```

@URCA decimal(7,3),
@WitD decimal(7,3),
@CHOL decimal(7,3),
@TGI decimal(7,3),
@HDL decimal(7,3),
@LDL decimal(7,3),
@oLDL decimal(7,3),
@LaboratoryjneInne varchar(1000),
@actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Wizyta where IdWizyta = @IdWizyta) )
    begin
        set @result = 3;
        set @message = 'visit of this ID =' + CAST(@IdWizyta as varchar) + ' not found';
        return;
    end;

    -- validators

    if dbo.validate_input_int('Wizyta', 'WynikWechu', @WynikWechu) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute WynikWechu - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'LimitDysfagii', @LimitDysfagii) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute LimitDysfagii - see enumeration for allowed values';
        return;
    end;

    if dbo.validate_input_int('Wizyta', 'USGWynik ', @USGWynik ) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute USGWynik - see enumeration for allowed values';
        return;
    end;

    update Wizyta
    set
        Holter = @Holter,
        BadanieWechu = @BadanieWechu,
        WynikWechu = @WynikWechu,
        LimitDysfagii = @LimitDysfagii,
        pH_metriaPrzełyku = @pH_metriaPrzełyku,
        SPECT = @SPECT,
        MRI = @MRI,
        MRIwynik = @MRIwynik,
        USGsrodmozgowia = @USGsrodmozgowia,
        USGWynik = @USGWynik,
        Genetyka = @Genetyka,
        GenetykaWynik = @GenetykaWynik,
        Surowica = @Surowica,
        SurowicaPozostało = @SurowicaPozostało,
        Ferrytyna = @Ferrytyna,
        CRP = @CRP,

```

```

        NTproCNP = @NTproCNP,
        URCA = @URCA,
        WitD = @WitD,
        CHOL = @CHOL,
        TGI = @TGI,
        HDL = @HDL,
        LDL = @LDL,
        oLDL = @oLDL,
        LaboratoryjneInne = @LaboratoryjneInne,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdWizyta = @IdWizyta;

    return;
end;
go

-- @result codes: 0 = OK, 1 = variant already exists while run in no-update mode, exist 2 = validation
-- failed - see message, 3 = visit of this ID not found, 4 = user login unknown
create procedure update_variant_examination_data_partA (@IdWizyta int, @DBS tinyint, @BMT bit,
    @UPDRS_I tinyint,
    @UPDRS_II tinyint,
    @UPDRS_18 tinyint,
    @UPDRS_19 tinyint,
    @UPDRS_20_FaceLipsChin tinyint,
    @UPDRS_20_RHand tinyint,
    @UPDRS_20_LHand tinyint,
    @UPDRS_20_RFoot tinyint,
    @UPDRS_20_LFoot tinyint,
    @UPDRS_21_RHand tinyint,
    @UPDRS_21_LHand tinyint,
    @UPDRS_22_Neck tinyint,
    @UPDRS_22_RHand tinyint,
    @UPDRS_22_LHand tinyint,
    @UPDRS_22_RFoot tinyint,
    @UPDRS_22_LFoot tinyint,
    @UPDRS_23_R tinyint,
    @UPDRS_23_L tinyint,
    @UPDRS_24_R tinyint,
    @UPDRS_24_L tinyint,
    @UPDRS_25_R tinyint,
    @UPDRS_25_L tinyint,
    @UPDRS_26_R tinyint,
    @UPDRS_26_L tinyint,
    @UPDRS_27 tinyint,
    @UPDRS_28 tinyint,
    @UPDRS_29 tinyint,
    @UPDRS_30 tinyint,
    @UPDRS_31 tinyint,
    @UPDRS_III tinyint,
    @UPDRS_IV tinyint,
    @UPDRS_TOTAL tinyint,
    @HYscale decimal(2,1),
    @SchwabEnglandScale tinyint,
    @JazzNovo bit,
    @Wideookulograf bit,
    @Latencymeter bit,
    @LatencymeterDurationLEFT decimal(6,2),
    @LatencymeterLatencyLEFT decimal(6,2),
    @LatencymeterAmplitudeLEFT decimal(6,2),
    @LatencymeterPeakVelocityLEFT decimal(6,2),
    @LatencymeterDurationRIGHT decimal(6,2),
    @LatencymeterLatencyRIGHT decimal(6,2),
    @LatencymeterAmplitudeRIGHT decimal(6,2),
    @LatencymeterPeakVelocityRIGHT decimal(6,2),
    @LatencymeterDurationALL decimal(6,2),
    @LatencymeterLatencyALL decimal(6,2),
    @LatencymeterAmplitudeALL decimal(6,2),
    @LatencymeterPeakVelocityALL decimal(6,2),
    @allow_update_existing bit,
    @actor_login varchar(50),
    @result int OUTPUT, @variant_id int OUTPUT, @message varchar(200) OUTPUT )

```



```

as
begin
    declare @user_id int;
    declare @update int;
    set @result = 0;
    set @message = '';
    set @update = 0;

    if( not exists (select * from Wizyta where IdWizyta = @IdWizyta) )
    begin
        set @result = 3;
        set @message = 'visit of this number not found';
        return;
    end;
    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if (select count(*) from Badanie where IdWizyta = @IdWizyta and DBS = @DBS and BMT = @BMT ) > 0
    begin
        if (@allow_update_existing = 0)
        begin
            set @result = 1;
            return;
        end;
        set @update = 1;
        select @variant_id = IdBadanie from Badanie where IdWizyta = @IdWizyta and DBS = @DBS and BMT = @BMT ✓
    ;
    end;

    -- validations

    -- TODO!
    if dbo.validate_input_int('Badanie', 'DBS', @DBS) = 0    begin    set @result = 2;    set @message =    ✓
    'Invalid value for attribute DBS';    return; end;

    if dbo.validate_input_int('Badanie', 'UPDRS_I', @UPDRS_I) = 0 begin set @result = 2; set @message =    ✓
    'Invalid value for attribute UPDRS_I'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_II', @UPDRS_II) = 0 begin set @result = 2; set @message =    ✓
    'Invalid value for attribute UPDRS_II'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_18', @UPDRS_18) = 0 begin set @result = 2; set @message =    ✓
    'Invalid value for attribute UPDRS_18'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_19', @UPDRS_19) = 0 begin set @result = 2; set @message =    ✓
    'Invalid value for attribute UPDRS_19'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_20_FaceLipsChin', @UPDRS_20_FaceLipsChin) = 0 begin set @    ✓
    result = 2; set @message = 'Invalid value for attribute UPDRS_20_FaceLipsChin'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_20_RHand', @UPDRS_20_RHand) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_20_RHand'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_20_LHand', @UPDRS_20_LHand) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_20_LHand'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_20_RFoot', @UPDRS_20_RFoot) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_20_RFoot'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_20_LFoot', @UPDRS_20_LFoot) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_20_LFoot'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_21_RHand', @UPDRS_21_RHand) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_21_RHand'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_21_LHand', @UPDRS_21_LHand) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_21_LHand'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_22_Neck', @UPDRS_22_Neck) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_22_Neck'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_22_RHand', @UPDRS_22_RHand) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_22_RHand'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_22_LHand', @UPDRS_22_LHand) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_22_LHand'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_22_RFoot', @UPDRS_22_RFoot) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_22_RFoot'; return; end;
    if dbo.validate_input_int('Badanie', 'UPDRS_22_LFoot', @UPDRS_22_LFoot) = 0 begin set @result = 2; set @    ✓
    message = 'Invalid value for attribute UPDRS_22_LFoot'; return; end;

```

```

if dbo.validate_input_int('Badanie', 'UPDRS_23_R', @UPDRS_23_R) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_23_R'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_23_L', @UPDRS_23_L) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_23_L'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_24_R', @UPDRS_24_R) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_24_R'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_24_L', @UPDRS_24_L) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_24_L'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_25_R', @UPDRS_25_R) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_25_R'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_25_L', @UPDRS_25_L) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_25_L'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_26_R', @UPDRS_26_R) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_26_R'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_26_L', @UPDRS_26_L) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_26_L'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_27', @UPDRS_27) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_27'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_28', @UPDRS_28) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_28'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_29', @UPDRS_29) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_29'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_30', @UPDRS_30) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_30'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_31', @UPDRS_31) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_31'; return; end;
if dbo.validate_input_int('Badanie', 'UPDRS_IV', @UPDRS_IV) = 0 begin set @result = 2; set @message = 'Invalid value for attribute UPDRS_IV'; return; end;

if dbo.validate_input_int('Badanie', 'SchwabEnglandScale', @SchwabEnglandScale) = 0 begin set @result = 2; set @message = 'Invalid value for attribute SchwabEnglandScale'; return; end;
if dbo.validate_input_decimal('Badanie', 'HYscale', @HYscale) = 0 begin set @result = 2; set @message = 'Invalid value for attribute HYscale'; return; end;

```

```

-- /validations
-- depending on update
if(@update = 0)
begin
insert into Badanie (
    IdWizyta,
    DBS,
    BMT,
    UPDRS_I,
    UPDRS_II,
    UPDRS_18,
    UPDRS_19 ,
    UPDRS_20_FaceLipsChin,
    UPDRS_20_RHand,
    UPDRS_20_LHand,
    UPDRS_20_RFoot,
    UPDRS_20_LFoot,
    UPDRS_21_RHand,
    UPDRS_21_LHand,
    UPDRS_22_Neck,
    UPDRS_22_RHand,
    UPDRS_22_LHand,
    UPDRS_22_RFoot,
    UPDRS_22_LFoot,
    UPDRS_23_R,
    UPDRS_23_L,
    UPDRS_24_R,
    UPDRS_24_L,
    UPDRS_25_R,
    UPDRS_25_L,
    UPDRS_26_R,
    UPDRS_26_L,
    UPDRS_27,
    UPDRS_28,
    UPDRS_29,
    UPDRS_30,
    UPDRS_31,
    UPDRS_III,
    UPDRS_IV,
    UPDRS_TOTAL,

```

```

        Hyscale,
        SchwabEnglandScale,
        JazzNovo,
        Wideookulograf,
        Latencymeter,
        LatencymeterDurationLEFT ,
        LatencymeterLatencyLEFT ,
        LatencymeterAmplitudeLEFT ,
        LatencymeterPeakVelocityLEFT ,
        LatencymeterDurationRIGHT ,
        LatencymeterLatencyRIGHT ,
        LatencymeterAmplitudeRIGHT ,
        LatencymeterPeakVelocityRIGHT ,
        LatencymeterDurationALL ,
        LatencymeterLatencyALL ,
        LatencymeterAmplitudeALL ,
        LatencymeterPeakVelocityALL,
        Wprowadzil,
        Zmodyfikowal,
        OstatniaZmiana
    )
values (
    @IdWizyta,
    @DBS,
    @BMT,
    @UPDRS_I,
    @UPDRS_II,
    @UPDRS_18,
    @UPDRS_19 ,
    @UPDRS_20_FaceLipsChin,
    @UPDRS_20_RHand,
    @UPDRS_20_LHand,
    @UPDRS_20_RFoot,
    @UPDRS_20_LFoot,
    @UPDRS_21_RHand,
    @UPDRS_21_LHand,
    @UPDRS_22_Neck,
    @UPDRS_22_RHand,
    @UPDRS_22_LHand,
    @UPDRS_22_RFoot,
    @UPDRS_22_LFoot,
    @UPDRS_23_R,
    @UPDRS_23_L,
    @UPDRS_24_R,
    @UPDRS_24_L,
    @UPDRS_25_R,
    @UPDRS_25_L,
    @UPDRS_26_R,
    @UPDRS_26_L,
    @UPDRS_27,
    @UPDRS_28,
    @UPDRS_29,
    @UPDRS_30,
    @UPDRS_31,
    @UPDRS_III,
    @UPDRS_IV,
    @UPDRS_TOTAL,
    @Hyscale,
    @SchwabEnglandScale,
    @JazzNovo,
    @Wideookulograf,
    @Latencymeter,
    @LatencymeterDurationLEFT ,
    @LatencymeterLatencyLEFT ,
    @LatencymeterAmplitudeLEFT ,
    @LatencymeterPeakVelocityLEFT ,
    @LatencymeterDurationRIGHT ,
    @LatencymeterLatencyRIGHT ,
    @LatencymeterAmplitudeRIGHT ,
    @LatencymeterPeakVelocityRIGHT ,
    @LatencymeterDurationALL ,
    @LatencymeterLatencyALL ,
    @LatencymeterAmplitudeALL ,
    @LatencymeterPeakVelocityALL,

```

```

        @user_id,
        @user_id,
        getdate()
    ) set @variant_id = SCOPE_IDENTITY();
end;
else
begin
    update Badanie
    set
        UPDRS_I = @UPDRS_I,
        UPDRS_II = @UPDRS_II,
        UPDRS_18 = @UPDRS_18,
        UPDRS_19 = @UPDRS_19,
        UPDRS_20_FaceLipsChin = @UPDRS_20_FaceLipsChin,
        UPDRS_20_RHand = @UPDRS_20_RHand,
        UPDRS_20_LHand = @UPDRS_20_LHand,
        UPDRS_20_RFoot = @UPDRS_20_RFoot,
        UPDRS_20_LFoot = @UPDRS_20_LFoot,
        UPDRS_21_RHand = @UPDRS_21_RHand,
        UPDRS_21_LHand = @UPDRS_21_LHand,
        UPDRS_22_Neck = @UPDRS_22_Neck,
        UPDRS_22_RHand = @UPDRS_22_RHand,
        UPDRS_22_LHand = @UPDRS_22_LHand,
        UPDRS_22_RFoot = @UPDRS_22_RFoot,
        UPDRS_22_LFoot = @UPDRS_22_LFoot,
        UPDRS_23_R = @UPDRS_23_R,
        UPDRS_23_L = @UPDRS_23_L,
        UPDRS_24_R = @UPDRS_24_R,
        UPDRS_24_L = @UPDRS_24_L,
        UPDRS_25_R = @UPDRS_25_R,
        UPDRS_25_L = @UPDRS_25_L,
        UPDRS_26_R = @UPDRS_26_R,
        UPDRS_26_L = @UPDRS_26_L,
        UPDRS_27 = @UPDRS_27,
        UPDRS_28 = @UPDRS_28,
        UPDRS_29 = @UPDRS_29,
        UPDRS_30 = @UPDRS_30,
        UPDRS_31 = @UPDRS_31,
        UPDRS_III = @UPDRS_III,
        UPDRS_IV = @UPDRS_IV,
        UPDRS_TOTAL = @UPDRS_TOTAL,
        Hyscale = @HYscale,
        SchwabEnglandScale = @SchwabEnglandScale,
        JazzNovo = @JazzNovo,
        Wideookulograf = @Wideookulograf,
        Latencymeter = @Latencymeter,
        LatencymeterDurationLEFT = @LatencymeterDurationLEFT ,
        LatencymeterLatencyLEFT = @LatencymeterLatencyLEFT ,
        LatencymeterAmplitudeLEFT = @LatencymeterAmplitudeLEFT ,
        LatencymeterPeakVelocityLEFT = @LatencymeterPeakVelocityLEFT ,
        LatencymeterDurationRIGHT = @LatencymeterDurationRIGHT ,
        LatencymeterLatencyRIGHT = @LatencymeterLatencyRIGHT ,
        LatencymeterAmplitudeRIGHT = @LatencymeterAmplitudeRIGHT ,
        LatencymeterPeakVelocityRIGHT = @LatencymeterPeakVelocityRIGHT ,
        LatencymeterDurationALL = @LatencymeterDurationALL ,
        LatencymeterLatencyALL = @LatencymeterLatencyALL ,
        LatencymeterAmplitudeALL = @LatencymeterAmplitudeALL ,
        LatencymeterPeakVelocityALL = @LatencymeterPeakVelocityALL,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdWizyta = @IdWizyta and DBS = @DBS and BMT = @BMT ;
end;
return;
end;
go

--
-- @result codes: 0 = OK, 3 = variant of this ID not found, exist 2 = validation failed - see message, 4 = ✖
-- user login unknown
create procedure update_variant_examination_data_partB ( @IdBadanie int,
    @Tremorometria bit,
    @TremorometriaLEFT_0_1 decimal(7,2),

```

```

    @TremorometriaLEFT_1_2 decimal(7,2),
    @TremorometriaLEFT_2_3 decimal(7,2),
    @TremorometriaLEFT_3_4 decimal(7,2),
    @TremorometriaLEFT_4_5 decimal(7,2),
    @TremorometriaLEFT_5_6 decimal(7,2),
    @TremorometriaLEFT_6_7 decimal(7,2),
    @TremorometriaLEFT_7_8 decimal(7,2),
    @TremorometriaLEFT_8_9 decimal(7,2),
    @TremorometriaLEFT_9_10 decimal(7,2),
    @TremorometriaLEFT_23_24 decimal(7,2),
    @TremorometriaRIGHT_0_1 decimal(7,2),
    @TremorometriaRIGHT_1_2 decimal(7,2),
    @TremorometriaRIGHT_2_3 decimal(7,2),
    @TremorometriaRIGHT_3_4 decimal(7,2),
    @TremorometriaRIGHT_4_5 decimal(7,2),
    @TremorometriaRIGHT_5_6 decimal(7,2),
    @TremorometriaRIGHT_6_7 decimal(7,2),
    @TremorometriaRIGHT_7_8 decimal(7,2),
    @TremorometriaRIGHT_8_9 decimal(7,2),
    @TremorometriaRIGHT_9_10 decimal(7,2),
    @TremorometriaRIGHT_23_24 decimal(7,2),
    @TestSchodkowy bit,
    @TestSchodkowyCzas1 decimal(4,2),
    @TestSchodkowyCzas2 decimal(4,2),
    @TestMarszu bit,
    @TestMarszuCzas1 decimal(4,2),
    @TestMarszuCzas2 decimal(4,2),
    @Posturografia bit,
    @MotionAnalysis bit,
    @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Badanie where IdBadanie = @IdBadanie ) )
    begin
        set @result = 3;
        set @message = 'variant of this ID ='+ CAST(@IdBadanie as varchar)+' not found';
        return;
    end;

    update Badanie
    set
        Tremorometria = @Tremorometria,
        TremorometriaLEFT_0_1 = @TremorometriaLEFT_0_1,
        TremorometriaLEFT_1_2 = @TremorometriaLEFT_1_2,
        TremorometriaLEFT_2_3 = @TremorometriaLEFT_2_3,
        TremorometriaLEFT_3_4 = @TremorometriaLEFT_3_4,
        TremorometriaLEFT_4_5 = @TremorometriaLEFT_4_5,
        TremorometriaLEFT_5_6 = @TremorometriaLEFT_5_6,
        TremorometriaLEFT_6_7 = @TremorometriaLEFT_6_7,
        TremorometriaLEFT_7_8 = @TremorometriaLEFT_7_8,
        TremorometriaLEFT_8_9 = @TremorometriaLEFT_8_9,
        TremorometriaLEFT_9_10 = @TremorometriaLEFT_9_10,
        TremorometriaLEFT_23_24 = @TremorometriaLEFT_23_24,
        TremorometriaRIGHT_0_1 = @TremorometriaRIGHT_0_1,
        TremorometriaRIGHT_1_2 = @TremorometriaRIGHT_1_2,
        TremorometriaRIGHT_2_3 = @TremorometriaRIGHT_2_3,
        TremorometriaRIGHT_3_4 = @TremorometriaRIGHT_3_4,
        TremorometriaRIGHT_4_5 = @TremorometriaRIGHT_4_5,
        TremorometriaRIGHT_5_6 = @TremorometriaRIGHT_5_6,

```

```

        TremorometriaRIGHT_6_7 = @TremorometriaRIGHT_6_7,
        TremorometriaRIGHT_7_8 = @TremorometriaRIGHT_7_8,
        TremorometriaRIGHT_8_9 = @TremorometriaRIGHT_8_9,
        TremorometriaRIGHT_9_10 = @TremorometriaRIGHT_9_10,
        TremorometriaRIGHT_23_24 = @TremorometriaRIGHT_23_24,
        TestSchodkowy = @TestSchodkowy,
        TestSchodkowyCzas1 = @TestSchodkowyCzas1,
        TestSchodkowyCzas2 = @TestSchodkowyCzas2,
        TestMarszu = @TestMarszu,
        TestMarszuCzas1 = @TestMarszuCzas1,
        TestMarszuCzas2 = @TestMarszuCzas2,
        Posturografia = @Posturografia,
        MotionAnalysis = @MotionAnalysis,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdBadanie = @IdBadanie;

    return;
end;
go

-- @result codes: 0 = OK, 3 = variant of this ID not found, exist 2 = validation failed - see message, 4 = ✖
-- user login unknown
create procedure update_variant_examination_data_partC ( @IdBadanie int,
    @UpAndGo decimal(3,1),
    @UpAndGoLiczby decimal(3,1),
    @UpAndGoKubekPrawa decimal(3,1),
    @UpAndGoKubekLewa decimal(3,1),
    @TST decimal(3,1),
    @TandemPivot tinyint,
    @WTT decimal(3,1),
    @actor_login varchar(50), @result int OUTPUT, @message varchar(200) OUTPUT )
as
begin

    declare @user_id int;
    set @user_id = 0;

    set @result = 0;

    select @user_id = dbo.identify_user(@actor_login);
    if(@user_id = 0)
    begin
        set @result = 4;
        set @message = 'user of this login not found';
        return;
    end;

    if(not exists(select * from Badanie where IdBadanie = @IdBadanie ) )
    begin
        set @result = 3;
        set @message = 'variant of this ID =' + CAST(@IdBadanie as varchar) + ' not found';
        return;
    end;

    -- validation
    if dbo.validate_input_int('Badanie', 'TandemPivot', @TandemPivot) = 0
    begin
        set @result = 2;
        set @message = 'Invalid value for attribute TandemPivot - see enumeration for allowed values';
        return;
    end;

    update Badanie
    set
        UpAndGo = @UpAndGo,
        UpAndGoLiczby = @UpAndGoLiczby,
        UpAndGoKubekPrawa = @UpAndGoKubekPrawa,
        UpAndGoKubekLewa = @UpAndGoKubekLewa,
        TST = @TST,

```

```

        TandemPivot = @TandemPivot,
        WTT = @WTT,
        Zmodyfikowal = @user_id,
        OstatniaZmiana = getdate()
    where IdBadanie = @IdBadanie;

    return;
end;
go

-- atrybuty wielowartosciowe

--
create type IntValueListUdt as table
(
    Value int
)
go

create function get_exam_multi_values( @attr_name varchar(50), @IdWizyta int )
returns table
as
return
select Wartosc from WartoscAtrybutuWizytyInt waw join Atrybut a on waw.IdAtrybut = a.IdAtrybut where
    IdWizyta = @IdWizyta and a.Nazwa = @attr_name;
go

--
-- 1 = attribute name unknown; 2 = exam_id not found; 3 = validation of value failed;
create procedure update_exam_int_attribute ( @IdWizyta int, @attr_name varchar(50), @values as
    IntValueListUdt readonly, @actor_login varchar(50), @result int OUTPUT )
as
begin
    declare @attr_id int;
    set @result = 0;
    set @attr_id = 0;

    select @attr_id = IdAtrybut from Atrybut where Tabela = 'Wizyta' and Nazwa = @attr_name;

    -- validation
    if @attr_id = 0
    begin
        set @result = 1;
        return;
    end;

    if not exists (select * from Wizyta where IdWizyta = @IdWizyta )
    begin
        set @result = 2;
        return;
    end;

    if exists ( select * from @values where dbo.validate_input_int('Wizyta', @attr_name, Value) = 0 )
    begin
        set @result = 3;
        return;
    end;
    delete from WartoscAtrybutuWizytyInt where IdWizyta = @IdWizyta and IdAtrybut = @attr_id and Wartosc not
        in (select Value from @values );

    insert into WartoscAtrybutuWizytyInt (IdWizyta, IdAtrybut, Wartosc) select @IdWizyta, @attr_id, Value
        from
            @values where Value not in (select Wartosc v from WartoscAtrybutuWizytyInt where IdWizyta = @
                IdWizyta and IdAtrybut = @attr_id) ;

end;
go

```

```
--
create function get_variant_multi_values( @attr_name varchar(50), @IdBadanie int )
returns table
as
return
select Wartosc from WartoscAtrybutuBadaniaInt where IdBadanie = @IdBadanie;
go

--
-- 1 = attribute name unknown; 2 = exam_id not found
create procedure update_variant_int_attribute ( @IdBadanie int, @attr_name varchar(50), @values as IntValueListUdt readonly, @actor_login varchar(50), @result int OUTPUT )
as
begin
    declare @attr_id int;
    set @attr_id = 0;

    select @attr_id = IdAtrybut from Atrybut where Tabela = 'Badanie' and Nazwa = @attr_name;

    -- validation
    if @attr_id = 0
    begin
        set @result = 1;
        return;
    end;

    if not exists (select * from Badanie where IdBadanie = @IdBadanie )
    begin
        set @result = 2;
        return;
    end;

    delete from WartoscAtrybutuBadaniaInt where IdBadanie = @IdBadanie and IdAtrybut = @attr_id and Wartosc
    not in (select Value from @values );

    insert into WartoscAtrybutuBadaniaInt (IdBadanie, IdAtrybut, Wartosc) select @IdBadanie, @attr_id,
    ( (select Value from @values) except (select Wartosc from WartoscAtrybutuBadaniaInt where IdBadanie
    = @IdBadanie and IdAtrybut = @attr_id) ) ;

end;
go

-- pobieranie plaskiej kopii bazy
-- =====

--
create view AtrybutyWielowartoscioweWizyty as (
SELECT  t.IdWizyta IdWizyta, a.Nazwa NazwaAtrybutu,
        stuff(
            (
                select  ',' + cast(t1.Wartosc as Varchar)
                from      WartoscAtrybutuWizytyInt t1
                where     t1.IdWizyta = t.IdWizyta and t1.IdAtrybut = t.IdAtrybut
                order by t1.Wartosc
                for xml path('')
            ),1,1,') Wartosci
FROM      WartoscAtrybutuWizytyInt t join Atrybut a on t.IdAtrybut = a.IdAtrybut
GROUP BY t.IdWizyta, t.IdAtrybut, a.Nazwa
)
go

-- updated: 2013-11-21
create procedure get_database_copy
as
SELECT
/* PACJENT */
    P.[NumerPacjenta]
  ,P.[RokUrodzenia]
  ,P.[MiesiacUrodzenia]
  ,P.[Plec]
  ,P.[Lokalizacja]
```



```

        ,P.[LiczbaElektrod]
        ,P.[NazwaGrupy]
/*      ,P.[Wprowadzil] as WprowadzilDanePacjenta
        ,P.[Zmodyfikowal] as KorygowalDanePacjenta
        ,P.[OstatniaZmiana] as OstatniaKorektaDanychPacjenta */
/* WIZYTA */
        ,W.[RodzajWizyty]
        ,W.[DataPrzyjecia]
        ,W.[DataWypisu]
        ,W.[MasaCiala]
        ,W.[DataOperacji]
        ,W.[Wyksztalcenie]
        ,W.[Rodzinnosc]
        ,W.[RokZachorowania]
        ,W.[MiesiacZachorowania]
        ,W.[RokBadania]
        ,W.[MiesiacBadania]
        ,W.[PierwszyObjaw]
        ,W.[Drzenie]
        ,W.[Sztynosc]
        ,W.[Spowolnienie]
        ,W.[ObjawyInne]
        ,W.[ObjawyInneJakie]
        ,W.[CzasOdPoczObjDowlLDopy]
        ,W.[DyskinezyObecnie]
        ,W.[DyskinezyOdLat]
        ,W.[FluktuacjeObecnie]
        ,W.[FluktuacjeOdLat]
        ,W.[CzasDyskinez]
        ,W.[CzasOFF]
        ,W.[PoprawaPoLDopie]
        ,W.[PrzebyteLeczenieOperacyjnePD]
        ,W.[Papierosy]
        ,W.[Kawa]
        ,W.[ZielonaHerbata]
        ,W.[Alkohol]
        ,W.[ZabiegowWZnieczOgPrzedRozpoznaniemPD]
        ,W.[Zamieszkanie]
        ,(select Wartosci from AtrybutyWielowartoscioweWizyty where NazwaAtrybutu = 'NarazenieNaToks' and
IdWizyta = W.IdWizyta) as NarazeniaNaToks
        ,W.[Uwagi]
        ,W.[Nadcisnienie]
        ,W.[BlokeryKanWapn]
        ,W.[DominujacyObjawObecnie]
        ,W.[DominujacyObjawUwagi]
        ,(select Wartosci from AtrybutyWielowartoscioweWizyty where NazwaAtrybutu = 'ObjawyAutonomiczne' and
IdWizyta = W.IdWizyta) as ObjawyAutonomiczne
        ,W.[RLS]
        ,W.[ObjawyPsychotyczne]
        ,W.[Depresja]
        ,W.[Otepienie]
        ,W.[Dyzartria]
        ,W.[DysfagiaObjaw]
        ,W.[RBD]
        ,W.[ZaburzenieRuchomosciGalekOcznych]
        ,W.[Apraksja]
        ,W.[TestKlaskania]
        ,W.[ZaburzeniaWechowe]
        ,W.[Ldopa]
        ,W.[LDopaObecnie]
        ,W.[Agonista]
        ,W.[AgonistaObecnie]
        ,W.[Amantadyna]
        ,W.[AmantadynaObecnie]
        ,W.[MAOBinh]
        ,W.[MAOBinhObecnie]
        ,W.[COMTinh]
        ,W.[COMTinhObecnie]
        ,W.[Cholinolityk]
        ,W.[CholinolitykObecnie]
        ,W.[LekiInne]
        ,W.[LekiInneJakie]
        ,W.[L_STIMOpis]
        ,W.[L_STIMAmplitude]

```

```
,W.[L_STIMDuration]
,W.[L_STIMFrequency]
,W.[R_STIMOpis]
,W.[R_STIMAmplitude]
,W.[R_STIMDuration]
,W.[R_STIMFrequency]
,W.[Wypis_Ldopa]
,W.[Wypis_LDopaObecnie]
,W.[Wypis_Agonista]
,W.[Wypis_AgonistaObecnie]
,W.[Wypis_Amantadyna]
,W.[Wypis_AmantadynaObecnie]
,W.[Wypis_MAOBinh]
,W.[Wypis_MAOBinhObecnie]
,W.[Wypis_COMTinh]
,W.[Wypis_COMTinhObecnie]
,W.[Wypis_Cholinolityk]
,W.[Wypis_CholinolitykObecnie]
,W.[Wypis_LekiInne]
,W.[Wypis_LekiInneJakie]
,W.[Wypis_L_STIMOpis]
,W.[Wypis_L_STIMAmplitude]
,W.[Wypis_L_STIMDuration]
,W.[Wypis_L_STIMFrequency]
,W.[Wypis_R_STIMOpis]
,W.[Wypis_R_STIMAmplitude]
,W.[Wypis_R_STIMDuration]
,W.[Wypis_R_STIMFrequency]
,W.[WydzielanieSliny]
,W.[Dysfagia]
,W.[DysfagiaCzestotliwosc]
,W.[Nudnosci]
,W.[Zaparcia]
,W.[TrudnosciWoddawaniuMoczu]
,W.[PotrzebaNagleOddaniaMoczu]
,W.[NiekompletneOproznieniePecherza]
,W.[SlabyStrumienMoczu]
,W.[CzestotliwoscOddawaniaMoczu]
,W.[Nykturia]
,W.[NiekontrolowaneOddawanieMoczu]
,W.[Omdlenia]
,W.[ZaburzeniaRytmuSerca]
,W.[ProbaPionizacyjna]
,W.[WzrostPodtlivostiTwarzKark]
,W.[WzrostPotliwosciRamionaDlonie]
,W.[WzrostPotliwosciBrzuchPlecy]
,W.[WzrostPotliwosciKonczynyDolneStopy]
,W.[SpadekPodtlivostiTwarzKark]
,W.[SpadekPotliwosciRamionaDlonie]
,W.[SpadekPotliwosciBrzuchPlecy]
,W.[SpadekPotliwosciKonczynyDolneStopy]
,W.[NietolerancjaWysokichTemp]
,W.[NietolerancjaNiskichTemp]
,W.[Lojotok]
,W.[SpadekLibido]
,W.[KlopotyOsiagnieciaErekcji]
,W.[KlopotyUtrzymaniaErekcji]
/* BADANIE (wariantowe) */
,B.[DBS]
,B.[BMT]
,B.[UPDRS_I]
,B.[UPDRS_II]
,B.[UPDRS_18]
,B.[UPDRS_19]
,B.[UPDRS_20_FaceLipsChin]
,B.[UPDRS_20_RHand]
,B.[UPDRS_20_LHand]
,B.[UPDRS_20_RFoot]
,B.[UPDRS_20_LFoot]
,B.[UPDRS_21_RHand]
,B.[UPDRS_21_LHand]
,B.[UPDRS_22_Neck]
,B.[UPDRS_22_RHand]
,B.[UPDRS_22_LHand]
```

```
,B.[UPDRS_22_RFoot]
,B.[UPDRS_22_LFoot]
,B.[UPDRS_23_R]
,B.[UPDRS_23_L]
,B.[UPDRS_24_R]
,B.[UPDRS_24_L]
,B.[UPDRS_25_R]
,B.[UPDRS_25_L]
,B.[UPDRS_26_R]
,B.[UPDRS_26_L]
,B.[UPDRS_27]
,B.[UPDRS_28]
,B.[UPDRS_29]
,B.[UPDRS_30]
,B.[UPDRS_31]
,B.[UPDRS_III]
,B.[UPDRS_IV]
,B.[UPDRS_TOTAL]
,B.[HYScale]
,B.[SchwabEnglandScale]
,B.[JazzNovo]
,B.[Wideookulograf]
,B.[Latencymeter]
,B.[LatencymeterDurationLEFT]
,B.[LatencymeterLatencyLEFT]
,B.[LatencymeterAmplitudeLEFT]
,B.[LatencymeterPeakVelocityLEFT]
,B.[LatencymeterDurationRIGHT]
,B.[LatencymeterLatencyRIGHT]
,B.[LatencymeterAmplitudeRIGHT]
,B.[LatencymeterPeakVelocityRIGHT]
,B.[LatencymeterDurationALL]
,B.[LatencymeterLatencyALL]
,B.[LatencymeterAmplitudeALL]
,B.[LatencymeterPeakVelocityALL]
,B.[Tremorometria]
,B.[TremorometriaLEFT_0_1]
,B.[TremorometriaLEFT_1_2]
,B.[TremorometriaLEFT_2_3]
,B.[TremorometriaLEFT_3_4]
,B.[TremorometriaLEFT_4_5]
,B.[TremorometriaLEFT_5_6]
,B.[TremorometriaLEFT_6_7]
,B.[TremorometriaLEFT_7_8]
,B.[TremorometriaLEFT_8_9]
,B.[TremorometriaLEFT_9_10]
,B.[TremorometriaLEFT_23_24]
,B.[TremorometriaRIGHT_0_1]
,B.[TremorometriaRIGHT_1_2]
,B.[TremorometriaRIGHT_2_3]
,B.[TremorometriaRIGHT_3_4]
,B.[TremorometriaRIGHT_4_5]
,B.[TremorometriaRIGHT_5_6]
,B.[TremorometriaRIGHT_6_7]
,B.[TremorometriaRIGHT_7_8]
,B.[TremorometriaRIGHT_8_9]
,B.[TremorometriaRIGHT_9_10]
,B.[TremorometriaRIGHT_23_24]
,B.[TestSchodkowy]
,B.[TestSchodkowyCzas1]
,B.[TestSchodkowyCzas2]
,B.[TestMarszu]
,B.[TestMarszuCzas1]
,B.[TestMarszuCzas2]
,B.[Posturografia]
,B.[MotionAnalysis]
,B.[UpAndGo]
,B.[UpAndGoLiczby]
,B.[UpAndGoKubekPrawa]
,B.[UpAndGoKubekLewa]
,B.[TST]
,B.[TandemPivot]
,B.[WTT]
,B.[Wprowadzil] as WariantZapisa1
```

```

        ,B.[Zmodyfikowal] as WariantModyfikowal
        ,B.[OstatniaZmiana] as OstatniaEdycjaWariantu
/* WIZYTA - c.d. */
        ,W.[PDQ39]
        ,W.[AIMS]
        ,W.[Epworth]
        ,W.[CGI]
        ,W.[FSS]
        ,W.[TestZegara]
        ,W.[MMSE]
        ,W.[WAIS_R_Wiadomosci]
        ,W.[WAIS_R_PowtarzanieCyfr]
        ,W.[SkalaDepresjiBecka]
        ,W.[TestFluencjiZwierzeta]
        ,W.[TestFluencjiOstre]
        ,W.[TestFluencjiK]
        ,W.[TestLaczeniaPunktowA]
        ,W.[TestLaczeniaPunktowB]
        ,W.[TestUczeniaSlownoSluchowego]
        ,W.[TestStroopa]
        ,W.[TestMinnesota]
        ,W.[InnePsychologiczne]
        ,W.[OpisBadania]
        ,W.[Wnioski]
        ,W.[Holter]
        ,W.[BadanieWechu]
        ,W.[WynikWechu]
        ,W.[LimitDysfagii]
        ,W.[pH_metriaPrzełyku]
        ,W.[SPECT]
        ,(select Wartosci from AtrybutyWielowartoscioweWizyty where NazwaAtrybutu = 'SPECTWynik' and IdWizyta =
        = W.IdWizyta) as SPECTWyniki
        ,W.[MRI]
        ,W.[MRIwynik]
        ,W.[USGsródmozgowia]
        ,W.[USGWynik]
        ,W.[Genetyka]
        ,W.[GenetykaWynik]
        ,W.[Surowica]
        ,W.[SurowicaPozostało]
        ,W.[Ferrytyna]
        ,W.[CRP]
        ,W.[NTproCNP]
        ,W.[URCA]
        ,W.[WitD]
        ,W.[CHOL]
        ,W.[TGI]
        ,W.[HDL]
        ,W.[LDL]
        ,W.[olLDL]
        ,W.[LaboratoryjneInne]
        ,W.[Wprowadzil] as WizyteWprowadzil
        ,W.[Zmodyfikowal] as WizyteEdytowal
        ,W.[OstatniaZmiana] as OstatniaModyfikacja

FROM Pacjent P left join Wizyta w on P.IdPacjent = W.IdPacjent left join Badanie B on B.IdWizyta = W.
        IdWizyta
order by P.NumerPacjenta, W.RodzajWizyty, B.BMT, B.DBS
go

```

```
-- wsparcie generowania numerow pacjenta
```

```

--
create procedure suggest_new_patient_number( @admission_date date, @patient_number varchar(20) OUTPUT )
as
begin
    declare @maxnum int;
    set @patient_number = '/PD/DBS/'+CAST(YEAR(@admission_date) as varchar);
    select @maxnum = max(IdPacjent)+1 from Pacjent;
    if @maxnum is null

```

```
        set @maxnum = 1;
        set @patient_number = cast(@maxnum as varchar)+@patient_number; -- rozwiązanie tymczasowe
        return;
    end;
go

--
create procedure suggest_new_patient_number_with_group( @admission_date date, @patient_group varchar(3), @
    patient_number varchar(20) OUTPUT )
as
begin
    declare @maxnum int;
    set @patient_number = '/PD/'+@patient_group+'/' + CAST(YEAR(@admission_date) as varchar);
    select @maxnum = max(IdPacjent)+1 from Pacjent;
    if @maxnum is null
        set @maxnum = 1;
    set @patient_number = cast(@maxnum as varchar)+@patient_number; -- rozwiązanie tymczasowe
    return;
end;
go
```