

## **PROJEKT BAZY DANYCH**

### **EWIDENCJA DEKLARACJI PODATNIKÓW „EDPPS”**

Projekt bazy danych urzędu skarbowego wykonany jako samodzielny projekt w ramach zajęć uczelnianych „Bazy danych”. Pomysł zaczerpnięty po odbytych praktykach w urzędzie skarbowym.

Celem bazy danych jest umożliwienie pracownikom Urzędu Skarbowego szybkiego i swobodnego dostępu do danych deklaracji składanych przez podatników niezbędnego do codziennego funkcjonowania urzędu.

Założenia bazy danych EDPPS:

- Chcemy przechowywać dane podatników (NIP, nazwa firmy, adres firmy, nr kontaktowy);
- Chcemy przechowywać unikatowe numery deklaracji;
- Chcemy przechowywać numery pomocnicze (mogą się powtarzać);
- Chcemy przechowywać datę złożenia i datę wpłynięcia deklaracji;
- Chcemy mieć możliwość wprowadzenia uwag odnośnie deklaracji;
- Chcemy przechowywać informacje o pracownikach urzędu skarbowego;
- Chcemy przechowywać informacje odnośnie stanowisk w urzędzie i pensjach;
- Chcemy wiedzieć, który pracownik zajmuje się daną deklaracją i jakimi deklaracjami zajmuje się pracownik;
- Chcemy wiedzieć czy w deklaracji jest kwota do wpłaty czy do zwrotu;
- Chcemy przechowywać informacje o rodzaju działalności podatnika;
- Chcemy znać decyzje wydane w sprawie podatników

Utworzyliśmy bazę danych, dodaliśmy przykładowe rekordy.

Tabela **git\_DECYZJE**:

NR_D	DECYZJA
0	BRAK DECYZJI
1	ZAWIADOMIENIE
2	PONAGLENIE
3	WEZWANIE
4	WYKREŚLENIE
5	PRZYWRÓCENIE
6	INFORMACJA
7	SPRAWDZENIE
8	ANALIZA
9	OBSERWACJA

Tabela **git\_DZIALALNOSC**:

ID_D	TYP	RODZAJ_DZIALALNOSCI
1	HANDLOWA	SP. Z O. O
2	GOSPODARCZA	SP. Z O. O
3	BUDOWLANA	S. A.
4	PRODUKCYJNA	SP. Z O. O
5	TRANSPORTOWA	SP. Z O. O
6	USŁUGOWA	S. A.
7	PRZEWÓZOWA	SP. Z O. O
8	WYTWÓRCZA	SP. Z O. O

Tabela **git\_FORMULARZ**:

NR	NR_POMOCNICZY	DATA_WPLYNIECIA	DATA_ZLOZENIA	ROK	MIESIAC	UWAGI	GIT_RODZAJE_ID_R	GIT_PODATNIK_ID_P	GIT_KWOTA_ID_KW
200510111	20200510111	05-MAY-20	04-MAY-20	2020	4	(null)	11	999	-1
200510222	20200510222	05-MAY-20	04-MAY-20	2020	4	(null)	11	998	-4
200510333	20200510333	05-MAY-20	04-MAY-20	2020	4	(null)	12	997	2
200215111	20200215111	11-FEB-20	04-FEB-20	2020	1	(null)	21	996	0
200215222	20200215222	11-FEB-20	04-FEB-20	2020	1	(null)	21	995	0
200215333	20200215333	11-FEB-20	04-FEB-20	2020	1	(null)	21	994	0
191205111	20191205111	02-DEC-19	29-NOV-19	2019	11	(null)	11	993	4
191205222	20191205222	02-DEC-19	29-NOV-19	2019	11	(null)	12	992	3
191205333	20191205333	02-DEC-19	29-NOV-19	2019	11	(null)	11	991	-2
191111111	20191111111	07-NOV-19	05-NOV-19	2019	10	(null)	11	994	3
191111222	20191111222	07-NOV-19	05-NOV-19	2019	10	(null)	12	995	3
191111333	20191111333	07-NOV-19	05-NOV-19	2019	10	(null)	11	998	-3
200412111	20200412111	08-APR-20	02-APR-20	2020	3	(null)	21	990	0
191205444	20191205444	02-DEC-19	29-NOV-19	2019	11	(null)	31	990	0
200412222	20200412111	08-APR-20	02-APR-20	2020	3	(null)	32	999	0

Tabela **git\_KWOTA**:

I...	RODZAJ_KWOTY	KWOTA
-4	ZWROT	1000000
-3	ZWROT	500000
-2	ZWROT	100000
-1	ZWROT	10000
1	WPLATA	10000
2	WPLATA	100000
3	WPLATA	500000
4	WPLATA	1000000
0	BRAK	0

Tabela **git\_MIASTA**:

...	MIASTO
00-000	Warszawa
30-000	Kraków
90-000	Łódź
50-000	Wrocław
60-000	Poznań
80-000	Gdańsk
70-000	Szczecin
85-000	Bydgoszcz
20-000	Lublin
15-000	Białystok

Tabela **git\_PODATNIK**:

ID_P	NAZWA	NIP	GIT_MIASTA_KOD	ULICA	NR	TELEFON	GIT_DZIALALNOSC_ID_D
999	GREGAM	10001234567	00-000	Tkaczyka	12	111222333	1
998	FASTA	20001234567	15-000	Perefa	43	222333444	5
997	POLBRAK	30001234567	80-000	Kopalniana	1	333444555	3
996	DSSTUDIO	40001234567	20-000	Ogródkowa	24	444555666	4
995	AREC	50001234567	85-000	Piaskowa	64	555666777	2
994	VITALI	60001234567	70-000	Kopenhage	23	666777888	7
993	FORTUNA	70001234567	60-000	Amelii	122	777888999	7
992	PARAFKA	80001234567	50-000	Franczka	34	888999111	8
991	KOPEC	90001234567	90-000	Portowa	25	999111222	1
990	KAPA	10011234567	30-000	Długa	2	111999111	6

Tabela **git\_POZYCJE**:

GIT_URZEDNIK_ID_U	GIT_FORMULARZ_NR
4	200215111
4	200215222
5	191205444
5	200412222
7	191111222
9	200510111
9	200510222
10	200510333
11	191111111
11	191205111
11	191205333
12	191111333
12	191205222
13	200215333
15	200412111

Tabela **git\_PRZYPISANIE**:

GIT_PODATNIK_ID_P	GIT_DECYZJE_NR_D
990	9
991	2
992	3
993	6
994	6
995	1
996	1
997	0
998	0
999	0

Tabela **git\_RODZAJE**:

ID_R	NAZWA
11	VAT-7
12	VAT-7K
21	CIT-8
22	CIT-8/O
23	CIT-5
24	CIT-7
31	NIP-1
32	NIP-2
33	NIP-5
34	ZAP-3

Tabela **git\_STANOWISKO**:

ID_STAN	STANOWISKO	PENSJA	OPIS
100	Naczelnik	5000	Naczelnik urzędu skarbowego
101	Zastępca	4000	Zastępca naczelnika urzędu skarbowego
102	Kierownik	3500	Kierownik działu / referatu
103	Informatyk	3500	Informatyk urzędu skarbowego
104	Kontroler	3300	Kontroler rozliczeń
105	Referent	2900	Starszy referent
106	Księgowy	2900	Starszy księgowy
107	Radca	3800	Radca prawny
108	Komornik	3000	(null)
109	Inspektor	2800	Inspektor ds. bezpieczeństwa

Tabela **git\_URZEDNIK**:

IMIE	NAZWISKO	GIT_MIASTA_KOD	ULICA	NR	TELEFON	GIT_STANOWISKO_ID_STAN
1 Jan	Kopasiuk	50-000	Długa	2	987654321	100
2 Jerzy	Tereczek	50-000	Kopalniana	15	987321654	101
3 Władysław	Derda	50-000	Florenca	32	987987987	105
4 Artur	Pomaska	50-000	Kwiatkowa	12	741852963	106
5 Emilia	Reclaw	50-000	Akacyjowa	3	741963852	102
6 Maria	Parka	50-000	Kaszubska	5	741741741	101
7 Lucyna	Stencel	50-000	Osiedlowa	38	963852741	105
8 Elżbieta	Duda	50-000	Portowa	43	963741852	103
9 Martyna	Labuda	50-000	Tkacka	5	963963963	105
10 Szymon	Opierski	50-000	Świętopełka	9	852963741	105
11 Tomasz	Nastaly	50-000	Mickiewicza	84	852741963	105
12 Kamil	Rolbiecki	50-000	Słoneczna	52	852852852	105
13 Agnieszka	Szulc	50-000	Słowackiego	3	753421869	106
14 Joanna	Raca	50-000	Rogali	4	869571324	107
15 Patrycja	Kołodziejczyk	50-000	Strzelecka	6	623951847	106

Stworzymy teraz parę widoków ułatwiających przeglądanie danych zawartych w bazie.

Widok **ile\_miesiecy** przedstawia ile miesięcy upłynęło od czasu złożenia deklaracji.

```
create or replace view ile_miesiecy as
select DATA_ZLOZENIA, SYSDATE as DATA,
FLOOR(MONTHS_BETWEEN(SYSDATE, DATA_ZLOZENIA)) as MIESIACE
from git_FORMULARZ
order by miesiace desc;
select * from ile_miesiecy;
```

DATA_ZLOZENIA	DATA	MIESIACE
29-NOV-19	03-JUL-20	7
29-NOV-19	03-JUL-20	7
05-NOV-19	03-JUL-20	7
05-NOV-19	03-JUL-20	7
05-NOV-19	03-JUL-20	7
29-NOV-19	03-JUL-20	7
29-NOV-19	03-JUL-20	7
04-FEB-20	03-JUL-20	4
04-FEB-20	03-JUL-20	4
04-FEB-20	03-JUL-20	4
02-APR-20	03-JUL-20	3
02-APR-20	03-JUL-20	3
04-MAY-20	03-JUL-20	1
04-MAY-20	03-JUL-20	1
04-MAY-20	03-JUL-20	1

Widok **ile\_stanowisk** prezentuje ilość pracowników obejmujących dane stanowiska.

```
create or replace view ile_stanowisk as
select git_STANOWISKO_ID_STAN as nr_stanowiska, count(*) as ilosc
from git_URZEDNIK
group by git_STANOWISKO_ID_STAN
order by ilosc desc;
select * from ile_stanowisk;
```

NR_STANOWISKA	ILOSC
105	6
106	3
101	2
103	1
102	1
100	1
107	1

Widok **pensje\_informacje** zawiera informacje o pensjach pracowników, tj. jaka jest minimalna, a jaka maksymalna pensja oraz średnie pensji i łączna kwota przeznaczana na pensje pracowników.

```
create or replace view pensje_informacje as
select
    min(pensja) as pensja_min, max(pensja) as pensja_max,
    sum(pensja) as suma_pensji, trunc(avg(pensja),2) as pensja_srednia
from git_STANOWISKO
order by pensja_max;
select * from pensje_informacje;
```

PENSJA_MIN	PENSJA_MAX	SUMA_PENSJI	PENSJA_SREDNIA
2800	5000	34700	3470

Widok **ile\_rodzaj** ukazuje ilość deklaracji danego rodzaju.

```
create or replace view ile_rodzaj as
select
  substr(nazwa,1,3) as RODZAJ,
  count(*) as ILOSC
from git_RODZAJE
group by substr(nazwa,1,3)
order by ILOSC desc;
select * from ile_rodzaj;
```

RODZAJ	ILOSC
CIT	4
NIP	3
VAT	2
ZAP	1

Widok **jakie\_stan** prezentuje stanowiska, które zajmują się formularzami i których pensja jest niższa od średniej.

```
create or replace view jakie_stan as
select st.stanowisko, st.pensja, count(fr.nr) as ilosc
from git_FORMULARZ fr left join git_POZYCJE pz
on fr.nr=pz.git_FORMULARZ_NR left join
git_URZEDNIK ur on pz.git_URZEDNIK_ID_U=ur.ID_U left join
git_STANOWISKO st on ur.git_STANOWISKO_ID_STAN=st.ID_STAN
group by stanowisko, pensja
having pensja < (select avg(pensja) from git_STANOWISKO);
select * from jakie_stan;
```

STANO...	PENSJA	ILOSC
Referent	2900	9
Księgowy	2900	4

Widok **jacy\_pod** przedstawia podatników z maksymalną kwotą w deklaracji.

```
create or replace view jacy_pod as
select fr.nr, nazwa, kwota
from git_PODATNIK pd right join git_FORMULARZ fr
on pd.id_p=fr.git_PODATNIK_ID_P left join
git_KWOTA kw on fr.git_KWOTA_ID_KW=kw.ID_KW
group by fr.nr, id_p, nazwa, kwota
having kwota=(select max(kwota) from git_KWOTA);
select * from jacy_pod;
```

NR	NAZWA	KWOTA
200510222	FASTA	1000000
191205111	FORTUNA	1000000

Widok **urz\_pod** przedstawia podatników, którymi zajmują się urzędniczki o pensjach niższych od średniej.

```
create or replace view urz_pod as
select ur.imie, st.pensja, pd.nazwa as ilosc
from git_PODATNIK pd right join git_FORMULARZ fr on
pd.ID_P=fr.git_PODATNIK_ID_P left join git_POZYCJE pz
on fr.nr=pz.git_FORMULARZ_NR left join
git_URZEDNIK ur on pz.git_URZEDNIK_ID_U=ur.ID_U left join
git_STANOWISKO st on ur.git_STANOWISKO_ID_STAN=st.ID_STAN
group by imie, pensja, pd.nazwa
having substr(imie,-1,1)='a' and pensja < (select avg(pensja) from git_STANOWISKO);
select * from urz_pod;
```

IMIE	PENSJA	ILOSC
Martyna	2900	FASTA
Patrycja	2900	KAPA
Agnieszka	2900	VITALI
Martyna	2900	GREGAM
Lucyna	2900	AREC



Widok **deklaracje** prezentuje niekwartalne deklaracje VAT-7 podatników wraz z numerami tych deklaracji, które zostały złożone w 2019 roku.

```
create or replace view deklaracje as
select pd.nazwa as podatnik, fr.nr, rd.nazwa as deklaracja,
extract(year from DATA_ZLOZENIA) as rok
from git_PODATNIK pd right join git_FORMULARZ fr on
pd.ID_P=fr.git_PODATNIK_ID_P left join git_RODZAJE rd
on fr.git_RODZAJE_ID_R=rd.ID_R
group by pd.nazwa, fr.nr, rd.nazwa, extract(year from DATA_ZLOZENIA)
having substr(rd.nazwa,1,3)='VAT' and substr(rd.nazwa,-1,1)!='K' and
extract(year from DATA_ZLOZENIA)=2019;
select * from deklaracje;
```

PODATNIK	NR	DEKLARACJA	ROK
FORTUNA	191205111	VAT-7	2019
FASTA	191111333	VAT-7	2019
KOPEC	191205333	VAT-7	2019
VITALI	191111111	VAT-7	2019

Stworzymy również pewne funkcję, procedury i wyzwalacze usprawniające funkcjonowanie i sprawdzające poprawność danych w EDPPS.

Funkcja **sprawdz\_nip** sprawdza poprawność numerów NIP opierając się na algorytmie sprawdzania rzeczywistych numerów NIP z małą modyfikacją ilości cyfr.

```
create or replace function sprawdz_nip (nip number:=00000000000) return varchar2 as
poprawny varchar2(3);
cyfry_nip tablica_cyfr := tablica_cyfr();
pomnozone_cyfry tablica_cyfr := tablica_cyfr();
suma number:=0;
begin
  for i in 1..11 loop cyfry_nip.extend;
    cyfry_nip(i):=(mod(NIP,power(10,12-i))-mod(NIP,power(10,11-i)))/power(10,11-i);
  end loop;
  pomnozone_cyfry.extend(9);
  pomnozone_cyfry(1):=cyfry_nip(1)*6;
  pomnozone_cyfry(2):=cyfry_nip(2)*5;
  pomnozone_cyfry(3):=cyfry_nip(3)*7;
  pomnozone_cyfry(4):=cyfry_nip(4)*2;
  pomnozone_cyfry(5):=cyfry_nip(5)*3;
  pomnozone_cyfry(6):=cyfry_nip(6)*4;
  pomnozone_cyfry(7):=cyfry_nip(7)*5;
  pomnozone_cyfry(8):=cyfry_nip(8)*6;
  pomnozone_cyfry(9):=cyfry_nip(9)*7;
  for i in 1..9 loop
    suma:=suma+pomnozone_cyfry(i)+cyfry_nip(10)+cyfry_nip(11);
  end loop; if mod(suma,11)=10 then poprawny:='NIE';
  else poprawny:='TAK'; end if;
  return poprawny;
end sprawdz_nip;
```

NAZWA	NIP	CZY_POPRAWNY
GREGAM	10001234567	NIE
FASTA	20001234567	TAK
POLBRAK	30001234567	TAK
DSSTUDIO	40001234567	TAK
AREC	50001234567	TAK

Funkcja **podwyżki** będzie przypisywała odpowiednią kwotę podwyżki zależnie od płci, ilości dokumentów oraz wysokości dotychczasowej pensji na podstawie wcześniej utworzonego widoku **pracownicy\_info**, który zawiera informacje na ten temat.

```
create or replace function podwyżki (
plec varchar2,
ilosc int,
pensja number,
pensja_srednia number
) return number as
podwyżka number;
begin
if plec='K' and ilosc >5 and pensja<pensja_srednia then podwyżka:=pensja;
elsif plec='K' and (ilosc >5 or pensja<pensja_srednia) then podwyżka:=0.5*pensja;
elsif plec='K' and ilosc<=5 and pensja>pensja_srednia then podwyżka:=0.2*pensja;
elsif plec='M' and ilosc >5 and pensja<pensja_srednia then podwyżka:=0.9*pensja;
elsif plec='M' and (ilosc >5 or pensja<pensja_srednia) then podwyżka:=0.4*pensja;
elsif plec='M' and ilosc<=5 and pensja>pensja_srednia then podwyżka:=0.1*pensja;
end if;
return podwyżka;
end podwyżki;
```

```
create or replace view pracownicy_info as select imie,
case when substr(imie,-1,1)='a' then 'K'
else 'M' end as plec, pensja,
count(fr.nr) as ilosc
from git_FORMULARZ fr left join git_POZYCJE pz on
fr.nr=pz.git_FORMULARZ_NR left join git_URZEDNIK ur on
pz.git_URZEDNIK_ID_U=ur.ID_U left join git_STANOWISKO st
on ur.git_STANOWISKO_ID_STAN=st.ID_STAN
group by imie, pensja;
```

IMIE	PLEC	PENSJA	ILOSC	PENSJA_SREDNIA	PODWYŻKA
Lucyna	K	2900	1	3470	1450
Szymon	M	2900	1	3470	1160
Patrycja	K	2900	1	3470	1450
Kamil	M	2900	2	3470	1160
Artur	M	2900	2	3470	1160

Teraz stworzymy procedurę **dodaj\_urzednika**, która doda podane w argumencie dane do widoku **urzednik\_dane** zawierającego dane urzędników posortowane po nr ID.

```
create or replace view urzednik_dane as select id_u, imie,
nazwisko, "176528_MIASTA_KOD", ulica, nr, telefon, "176528_STANOWISKO_ID_STAN"
from "176528_URZEDNIK"
order by id_u;
```

```
create or replace procedure dodaj_urzednika(id int, imie varchar2,
nazwisko varchar2, kod varchar2, ulica varchar2, nr varchar2, telefon varchar2,
id_stan int) as
begin
if (id>15) and (id_stan in (100,101,102,103,104,105,106,107)) and (length(telefon)=9) and (kod in ('00-000',
'30-000', '90-000', '50-000', '60-000', '80-000', '70-000', '85-000', '20-000', '15-000')) then
insert into urzednik_dane values (id, upper(substr(imie,1,1))||lower(substr(imie,2,length(imie)-1)),
upper(substr(nazwisko,1,1))||lower(substr(nazwisko,2,length(nazwisko)-1)),
kod, upper(substr(ulica,1,1))||lower(substr(ulica,2,length(ulica)-1)), nr, telefon, id_stan);
end if;
execute dodaj_urzednika(16, 'Marek', 'Kopera', '50-000', 'Koperka', '23', '159857128', 105);
execute dodaj_urzednika(17, 'ARKADIUSZ', 'Gorora', '50-000', 'akacjOwa', '25', '258753951', 105);
execute dodaj_urzednika(18, 'WikTOR', 'WitKOWSKI', '50-000', 'bAbelKOwa', '2', '148625975', 105);
select * from urzednik_dane;
```

14	Joanna	Raca	50-000	Rogali	4	869571324	107
15	Patrycja	Kołodziejczyk	50-000	Strzelecka	6	623951847	106
16	Marek	Kopera	50-000	Koperka	23	159857128	105
17	Arkadiusz	Gorora	50-000	Akacjowa	25	258753951	105
18	Wiktor	Witkowski	50-000	Babelkowa	2	148625975	105

Wyzwalacz **dodaj\_podatnika** przed dodaniem nowego podatnika sprawdzi czy podany kod miasta i działalności jest prawidłowy, jeżeli tak nie jest to doda nowe miejsce w odpowiedniej tabeli.

```
create or replace trigger dodaj_podatnika before insert on git_PODATNIK
for each row
declare
kod varchar2(6) := :new.git_MIASTA_KOD;
id_d int := :new.git_DZIALALNOSC_ID_D;
tel varchar2(9) := :new.telefon;
begin
if kod not in ('00-000','30-000','90-000','50-000','60-000'
,'80-000','70-000','85-000','20-000','15-000') then
insert into git_MIASTA values (kod,'MIASTO');
elsif id_d not in (1,2,3,4,5,6,7,8) then
insert into git_DZIALALNOSC values (id_d, 'TYP','OPIS');
elsif length(kod) != 6 then raise_application_error(-205402, 'Kod musi miec 6 znakow!');
elsif length(tel) != 9 then raise_application_error(-250430, 'Numer telefonów musi miec 9 znaków!');
end if;
end;
```

85-000	Bydgoszcz
20-000	Lublin
15-000	Białystok
80-405	MIASTO

7 PRZEWOZOWA	SP. Z O. O
8 WYTWÓRCZA	SP. Z O. O
9 TYP	OPIS

Wyzwalacz **logi\_zmian** będzie zapisywał w nowej tabeli **logi\_zmian** odpowiednie informacje do odpowiednich działań, gdzie tabela ta przechowywać będzie informacje o zmianach dokonanych w tabeli **git\_PODATNIK**.

```
create sequence logi_seq_id start with 1 increment by 1 nomaxvalue;
create table logi_zmian(id int, stara_wartosc varchar2(50),
zmiana varchar2(15), nowa_wartosc varchar2(50), data_modyfikacji date);
```

```
create or replace trigger log_zmian after insert or update or delete on git_PODATNIK for each row
declare
s_podatnik varchar2(50) := :old.nazwa;
n_podatnik varchar2(50) := :new.nazwa;
begin
if inserting then insert into logi_zmian values (logi_seq_id.nextval,
s_podatnik,'INSERT', n_podatnik, trunc(sysdate));
elsif updating then insert into logi_zmian values (logi_seq_id.nextval,
s_podatnik,'UPDATE', n_podatnik, trunc(sysdate));
elsif deleting then insert into logi_zmian values (logi_seq_id.nextval,
s_podatnik,'DELETE', n_podatnik, trunc(sysdate));
end if;
end;
```

ID	STARA_WARTOSC	ZMIANA	NOWA_WARTOSC	DATA_MODYFIKACJI
1	(null)	INSERT	Passaka	03-JUL-20
2	Groham	UPDATE	BEP	03-JUL-20
3	Passaka	DELETE	(null)	03-JUL-20



Wyzwalacz **po\_urzedniku** podczas dodawania rekordów do widoku **urzednik\_dane\_szcz** doda odpowiednie wartości także do rzeczywistych tabel.

```
create or replace view urzednik_dane_szcz as select id_u, imie,
nazwisko, kod, miasto, ulica, nr, telefon, id_stan, stanowisko, pensja
from git_MIASTA mi right join git_URZEDNIK ur on mi.kod=ur.git_MIASTA_KOD left join
git_STANOWISKO st on ur.git_STANOWISKO_ID_STAN=st.ID_STAN
order by id_u;
```

```
create or replace trigger po_urzedniku instead of insert on urzednik_dane_szcz for each row
begin
insert into git_MIASTA values (:new.kod, upper(substr(:new.miasto,1,1))||lower(substr(:new.miasto,2,length(:new.miasto)-1)));
insert into git_STANOWISKO values (:new.id_stan, upper(substr(:new.stanowisko,1,1))||lower(substr(:new.stanowisko,2,length(:new.stanowisko)-1)),
:new.pensja, 'OPIS');
insert into git_URZEDNIK values (:new.id_u,
upper(substr(:new.imie,1,1))||lower(substr(:new.imie,2,length(:new.imie)-1)),
upper(substr(:new.nazwisko,1,1))||lower(substr(:new.nazwisko,2,length(:new.nazwisko)-1)),
:new.kod, upper(substr(:new.ulica,1,1))||lower(substr(:new.ulica,2,length(:new.ulica)-1)),
:new.nr, :new.telefon, :new.id_stan);
end;
```

```
insert into urzednik_dane_szcz values(19,'Andzela','Kazera','55-000','Amsterdam','Korka','3','198757128',115,'Stanowisko1',2500);
insert into urzednik_dane_szcz values(20,'MAGda','Grerora','23-000','Londyn','acjOwa','5','681753951',120,'stanowisko2',3000);
insert into urzednik_dane_szcz values(21,'HoracjA','WiaWSKA','50-123','WaszyngtoN','beLKOWa','2','175925975',134,'stanoWISKO3',3200);

select * from git_MIASTA;
select * from git_STANOWISKO;
select * from git_URZEDNIK;
select * from urzednik_dane_szcz;
```

18	Wiktoria	Witkowska	50-000	Wrocław	Babelkowa	2	148625975	105	Referent	2900
19	Andzela	Kazera	55-000	Amsterdam	Korka	3	198757128	115	Stanowisko1	2500
20	Magda	Grerora	23-000	Londyn	Acjowa	5	681753951	120	Stanowisko2	3000
21	Horacja	Wiawska	50-123	Waszyngton	Belkowa	2	175925975	134	Stanowisko3	3200

19	Andzela	Kazera	55-000	Korka	3	198757128	115
20	Magda	Grerora	23-000	Acjowa	5	681753951	120
21	Horacja	Wiawska	50-123	Belkowa	2	175925975	134

109	Inspektor	2800	Inspektor ds. bezpieczeństwa
115	Stanowisko1	2500	OPIS
120	Stanowisko2	3000	OPIS
134	Stanowisko3	3200	OPIS

80-405	MIASTO
55-000	Amsterdam
23-000	Londyn
50-123	Waszyngton