|  |  |
| --- | --- |
| **CREATE OR REPLACE FUNCTION** osebe\_identifikacija*(i\_uporabnik varchar, i\_geslo varchar)* **RETURNS int AS**  **$$**  **DECLARE**  rezultat int**;**  **BEGIN**  **SELECT** id\_osebe **INTO** rezultat **FROM** osebe **WHERE** i\_uporabnik = uporabnisko\_ime **AND** i\_geslo = osebe.geslo;  **RETURN** rezultat**;**  **END;**  **$$ LANGUAGE plpgsql;**  DROP FUNCTION osebe\_identifikacija(i\_uporabnik varchar, i\_geslo varchar);  **SELECT \* FROM** osebe\_identifikacija('cene123', 'test')**;** | Vse razumljivo |

|  |  |
| --- | --- |
| **CREATE OR REPLACE FUNCTION** osebe\_i( i\_ime varchar, i\_priimek varchar) **RETURNS** **int** **AS**  **$$**  **DECLARE**  kljuc int = nextval('osebe\_id\_osebe\_seq')**;**  **BEGIN**  **INSERT INTO** osebe(id\_osebe, ime, priimek) **VALUES**(kljuc, i\_ime, i\_priimek)**;**  **RETURN** kljuc**;**  **END;**  **$$ LANGUAGE plpgsql;**  **SELECT \* FROM** osebe\_i('Janez', 'Novak')**;** | **nextval**('osebe\_id\_osebe\_seq') **->** nasledni ID v tabeli |

|  |  |
| --- | --- |
| **CREATE OR REPLACE FUNCTION** osebe\_vd(i\_id\_osebe int) **RETURNS** **int** **AS**  **$$**  **DECLARE**  rezultat int;  **BEGIN**  **DELETE FROM** osebe **WHERE** id\_osebe = i\_id\_osebe**;**  **GET DIAGNOSTICS** rezultat = **ROW\_COUNT;**  **RETURN** rezultat**;**  **END;**  **$$ language plpgsql;**  **SELECT \* FROM** osebe\_vd(82)**;** | **GET DIAGNOSTICS ->** zabeleži karkoli se zgodi v funkciji **->** *v tem primeru izbrisanih tabel*  **ROW\_COUNT ->** beleži več vrstic hkrati |

|  |  |  |
| --- | --- | --- |
| **CREATE OR REPLACE FUNCTION** osebe\_v2d(i\_ime varchar, i\_priimek varchar) **RETURNS** **int** **AS**  **$$**  **DECLARE**  rezultat int;  **BEGIN**  **DELETE FROM** osebe **WHERE** ime = i\_ime AND priimek = i\_priimek**;**  **GET DIAGNOSTICS** rezultat = **ROW\_COUNT;**  **RETURN** rezultat**;**  **END;**  **$$ language plpgsql;**  **SELECT \* FROM** osebe\_v2d('Janez', 'Novak')**;** | **GET DIAGNOSTICS ->** zabeleži karkoli se zgodi v funkciji **->** *v tem primeru izbrisanih tabel*  **ROW\_COUNT ->** beleži več vrstic hkrati |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CREATE TYPE** **rezultat12** **AS** **(**  Priimek varchar,  Ime varchar,  Naslov varchar,  Posta varchar,  KrajMesto varchar  **);**  **CREATE OR REPLACE FUNCTION** osebe\_vs(i\_krajMesto varchar) **RETURNS SETOF** rezultat12 **AS**  **$$**  **DECLARE**  vrstica rezultat12;  **BEGIN**  **FOR** vrstica **IN SELECT** priimek, ime, naslov, postna\_stevilka, ps.kraj  **FROM** osebe **INNER JOIN** public.postne\_stevilke ps on osebe.id\_postne\_stevilke = ps.id\_postne\_stevilke  **WHERE UPPER**(ps.kraj) **LIKE UPPER**(i\_krajMesto) || '%'  **LOOP**  **RETURN NEXT** vrstica;  **END LOOP;**  **END;**  **$$ LANGUAGE plpgsql;**  **SELECT \* FROM** osebe\_vs('L'); | **CREATE TYPE** **->** Ustvari tabelo zapisov (več kot enega)  Npr.:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Priimek | Ime | Naslov | Posta | KrajMesto |   **RETURNS SETOF ->** Vrne celotno skupino zapisov, ki so navedeni v CREATE TYPE  Vrstica je *spremenljivka*, rezultat12 pa *podatkovni tip.*  **LOOP** -> loop :D |

Is NULL ... THEN

IS NOT NULL ... THEN

EXCEPTION -

THEN RAISE EXCEPTION

AND COALESCE

%TYPE

%ROWTYPE

CREATE ROLE (GRAND/ROLE)

CREATE SCHEMA

DEBUG, LOG, INFO

NOTICE, WARNING

'''

```

**CREATE OR REPLACE FUNCTION** ime\_funkcije *(1ime\_sprtemenljivke, 2ime\_sprtemenljivke)* **RETURNS NEKI AS**

**$$**

**DECLARE**

**;**

**BEGIN**

**RETURN ;**

**END;**

**$$ LANGUAGE plpgsql;**

**SELECT \* FROM;**

```

Table

Description automatically generated