

# **BAZE PODATAKA 2**

**Doc.dr.sc. GORAN KRALJEVIĆ**

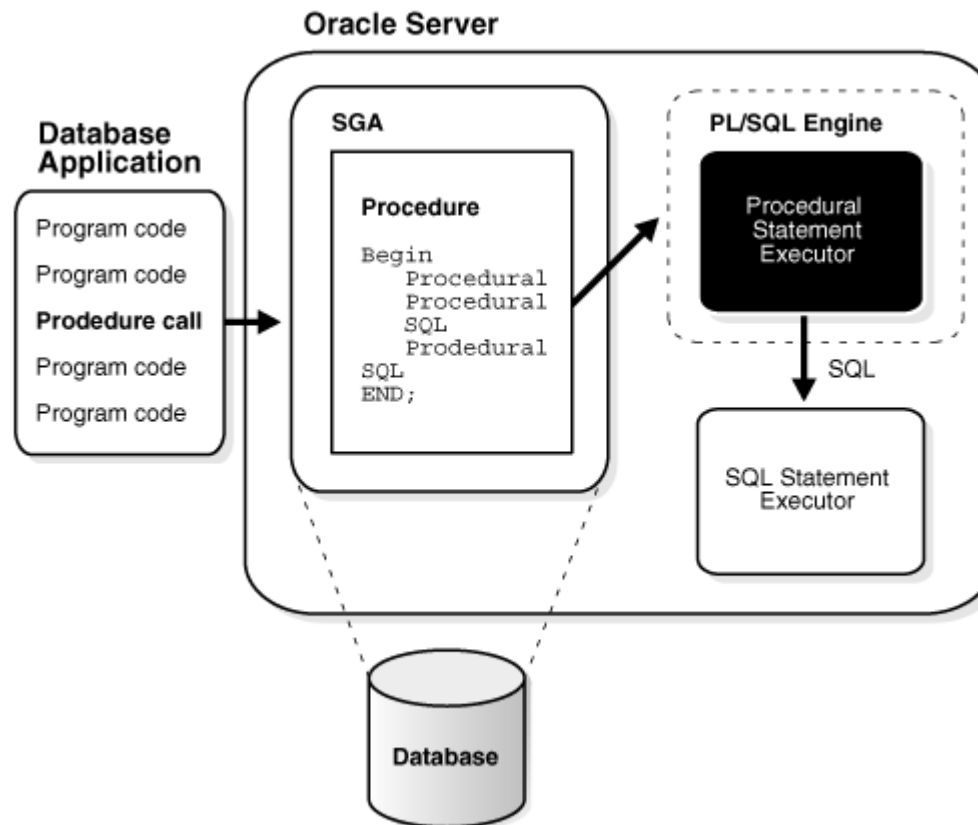
**Web:**  
**<http://www2.fsr.ba/nastava/baze>**

**Pitanja, primjedbe, dogovor za konzultacije ...**

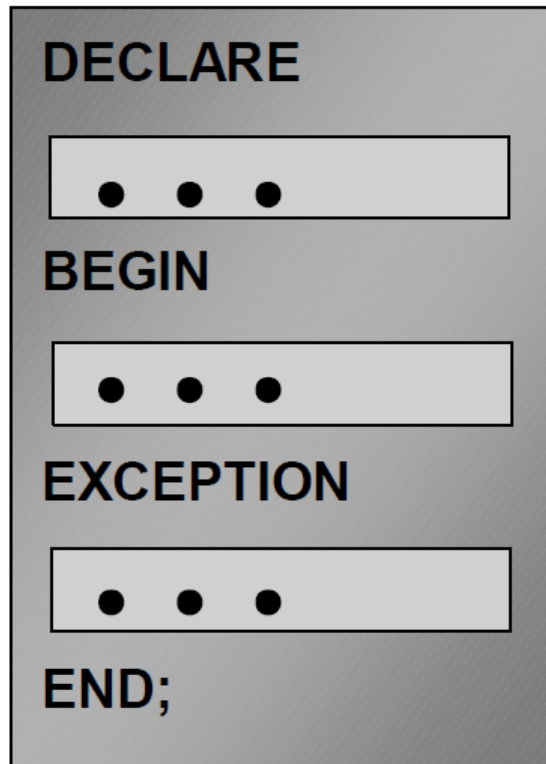
➤ E-mail: ***[goran.kraljevic@hteronet.ba](mailto:goran.kraljevic@hteronet.ba)***

# PL/SQL

- **Oracle: PL/SQL**  
*PL/SQL (Procedural Language / Structured Query Language)*



- **PL/SQL blok**



## **DECLARE**

v\_varijabla VARCHAR2(5);

## **BEGIN**

SELECT naziv\_kolone  
INTO v\_varijabla  
FROM naziv\_tablice;

## **EXCEPTION**

WHEN naziv\_iznimke THEN  
...

**END;**

Tipovi varijabli mogu biti definirani **eksplicitno** ili **implicitno**, prema tipovima atributa u relacijama baze podataka.

## DECLARE

```
v_ime VARCHAR2(25);  
v_prezime djelatnik.prezime%TYPE;  
v_datum_zaposlenja DATE := '01.01.2013';  
v_placa NUMBER(7,2) := 0;  
...
```

## BEGIN

```
...
```

## END;

SELECT izraz u PL/SQL-u:

**DECLARE**

v\_sum\_placa NUMBER(7,2);

**BEGIN**

**SELECT** SUM(placa)

**INTO** v\_sum\_placa

**FROM** djelatnik;

DBMS\_OUTPUT.PUT\_LINE ('Ukupna suma placa je: ' ||  
TO\_CHAR(v\_sum\_placa));

**END;**

- U PL/SQL bloku je moguće koristiti sljedeće atribute (u nekom uvjetu) za verificiranje rezultata DML izraza (SELECT, INSERT, UPDATE, DELETE)
  - **SQL%ROWCOUNT**
  - **SQL%FOUND**
  - **SQL%NOTFOUND**
  - **SQL%ISOPEN**



- PL/SQL – Kontrola programskog toka

**IF** uvjet **THEN**

...;

[**ELSIF** uvjet **THEN**

...;]

[**ELSE**

...;]

**END IF;**

**LOOP**

...;

...;

**EXIT** [**WHEN** uvjet];

**END LOOP;**

**WHILE** uvjet

**LOOP**

...;

...;

**END LOOP;**

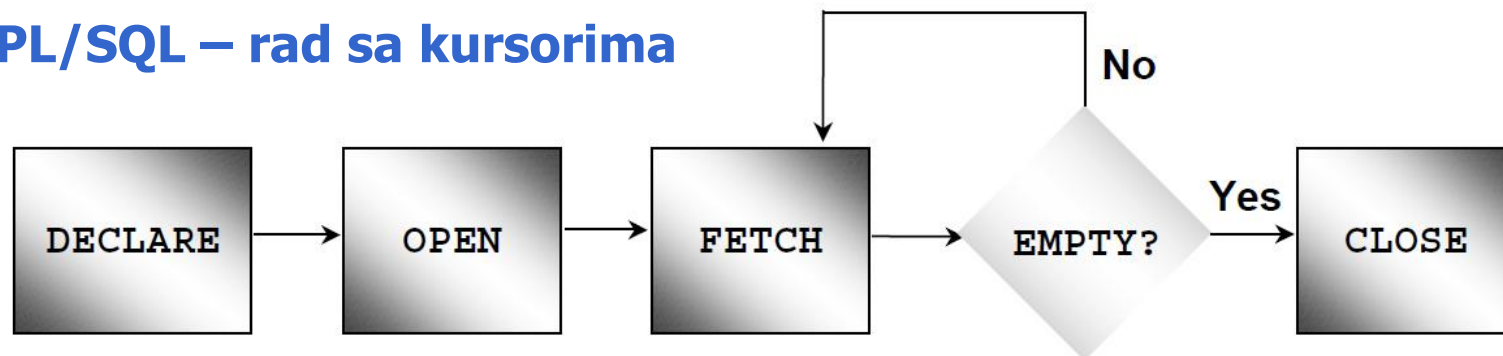
**FOR** brojac **IN** od..do **LOOP**

...;

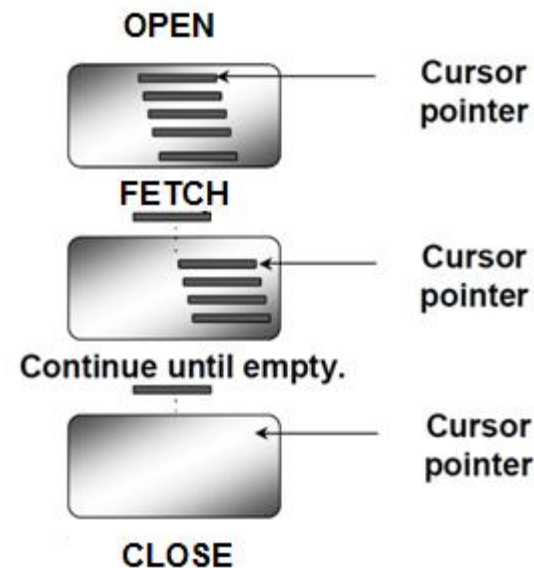
...;

**END LOOP;**

- PL/SQL – rad sa kursorima



```
DECLARE  
CURSOR naziv_kursora IS  
  SELECT ... ;  
BEGIN  
  OPEN naziv_kursora;  
  LOOP  
    FETCH naziv_kursora INTO ... ;  
    EXIT WHEN ... ;  
  END LOOP;  
  CLOSE naziv_kursora;  
END ;
```



- Rukovanje iznimkama (exceptions):

```
EXCEPTION  
  WHEN iznimka1 [OR iznimka2 ...] THEN  
    ...;  
    ...;  
  [WHEN iznimka3 THEN  
    ...;  
    ...; ]  
  [WHEN OTHERS THEN  
    ...;  
    ...; ]
```

- Primjeri predefiniranih iznimki:

**NO\_DATA\_FOUND, TOO\_MANY\_ROWS, ZERO\_DIVIDE**

- **RAISE\_APPLICATION\_ERROR**

```
BEGIN
    ... SQL izraz ...
    IF SQL%NOTFOUND THEN
        RAISE_APPLICATION_ERROR (-20000, 'Greška...');
    END IF;
    ...
```

```
...
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR (-20000, 'Greška...');
END;
```

## „Anonimni“ blok:

```
[DECLARE]  
BEGIN  
    ...;  
[EXCEPTION]  
END;
```

## Procedura:

```
PROCEDURE naziv  
IS  
BEGIN  
    ...;  
[EXCEPTION]  
END;
```

## Funkcija:

```
FUNCTION naziv  
RETURN tip_pod  
IS  
BEGIN  
    ...;  
RETURN vrijednost;  
[EXCEPTION]  
END;
```

- Kreiranje pohranjene procedure:

```
CREATE [OR REPLACE] PROCEDURE naziv_procedure  
    [(parametar1 [IN|OUT|IN OUT] tip_podatka1,  
     parametar2 [IN|OUT|IN OUT] tip_podatka2,  
     . . .)]  
IS | AS  
    PL/SQL blok;
```

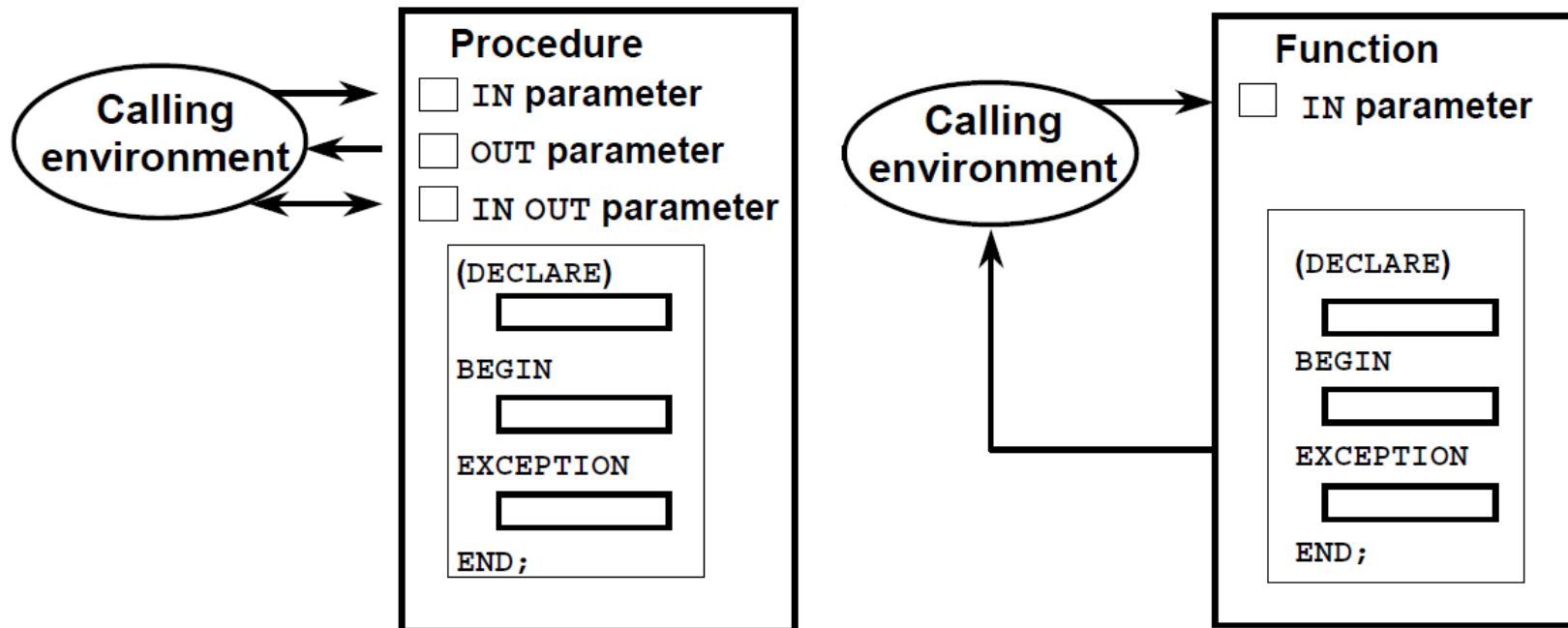
- Poziv pohranjene procedure:
  - **EXECUTE naziv\_procedure** (parametri...);
  - **naziv\_procedure** (parametri...);

- Kreiranje funkcije:

```
CREATE [OR REPLACE] FUNCTION naziv_funkcije  
    [(parametar1 [IN] tip_podatka1,  
     parametar2 [IN] tip_podatka2,  
     . . .)]  
RETURN tip_podatka  
IS | AS  
PL/SQL blok;
```

- Napomena:
  - PL/SQL blok mora imati najmanje jedan **RETURN** iskaz !

- Procedura ili funkcija ?





- Kreiranje okidača (trigera):

```
CREATE [OR REPLACE] TRIGGER naziv_okidaca  
    BEFORE | AFTER  
    INSERT | UPDATE (OF naziv_atributa) | DELETE  
    ON naziv_tablice  
    [FOR EACH ROW]  
    PL/SQL blok;
```

- Ukoliko je riječ o okidaču koji se aktivira UPDATE naredbom, moguće je definirati određene attribute unutar tablice čija promjena će aktivirati okidač (UPDATE OF naziv\_atributa).
- FOR EACH ROW – ako se navede ova opcija, akcije unutar okidača će se izvršiti za svaku n-torku na koju je djelovala operacija koja je aktivirala okidač.

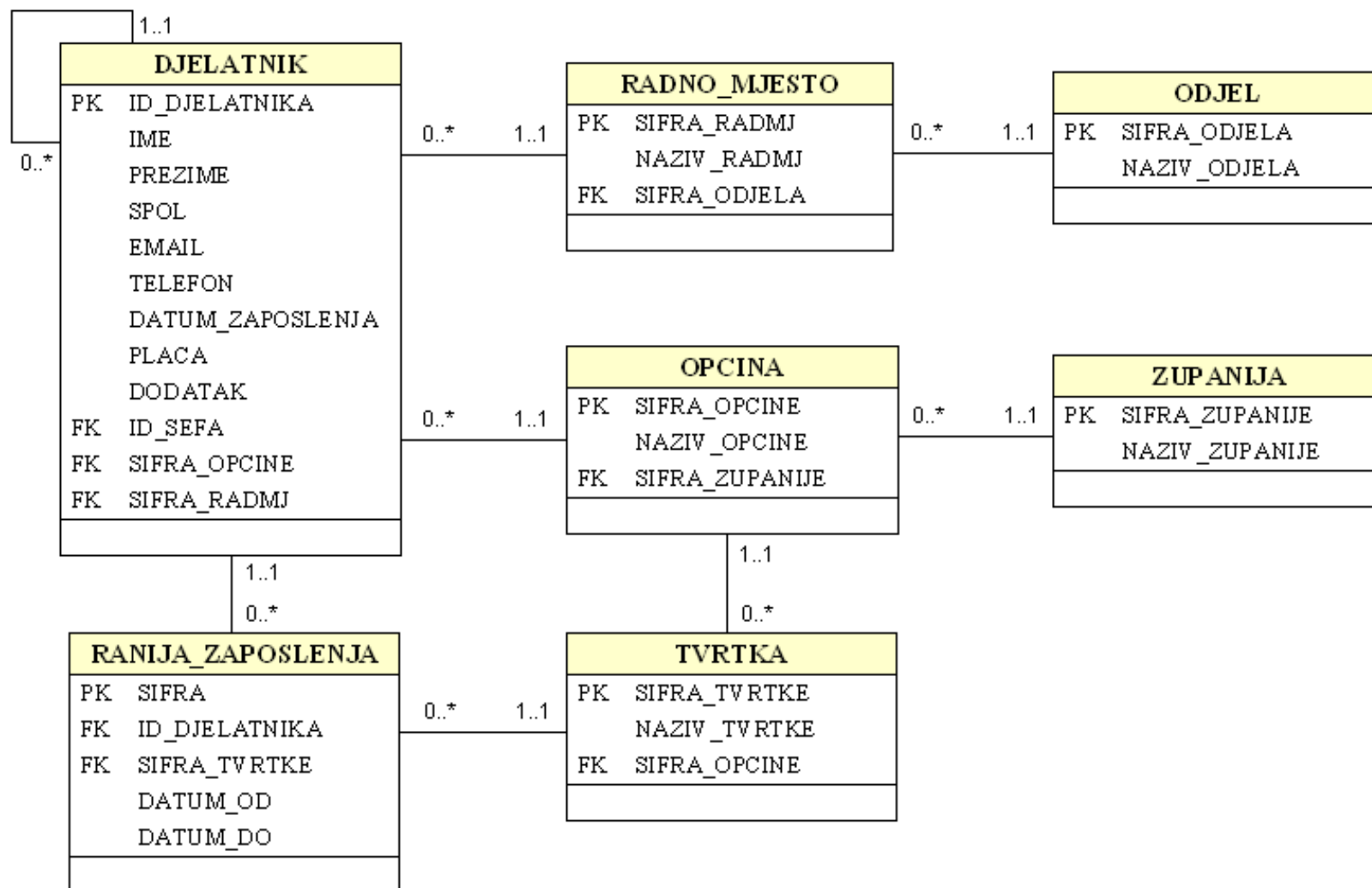
- U okidaču je moguće provjeriti (u nekom uvjetu) o kojoj akciji je riječ:
  - **INSERTING, UPDATING, DELETING**
- Ukoliko je okidač definiran na razini retka, moguće je koristiti referencu na vrijednosti u retku prije akcije (INSERT, UPDATE, DELETE) i poslije akcije:
  - **:OLD**
  - **:NEW**

DML naredba	Stara vrijednost (OLD)	Nova vrijednost (NEW)
INSERT	NULL	Upisana (insertirana) vrijednost
UPDATE	Vrijednost prije UPDATE naredbe	Vrijednost poslije UPDATE naredbe
DELETE	Vrijednost prije brisanja	NULL

# Primjeri

# Testna baza

(napunjena testnim podacima ...)



# PL/SQL – Primjer (kursor)

**DECLARE**

**CURSOR** djelatnik\_cursor IS

SELECT ime, prezime, placa, datum\_zaposlenja FROM djelatnik;

v\_ime VARCHAR2(25);

v\_prezime VARCHAR2(25);

v\_placa NUMBER(7,2);

v\_datum\_zaposlenja DATE;

**BEGIN**

**OPEN** djelatnik\_cursor;

LOOP

**FETCH** djelatnik\_cursor

INTO v\_ime, v\_prezime, v\_placa, v\_datum\_zaposlenja;

EXIT WHEN djelatnik\_cursor%NOTFOUND;

IF v\_placa > 1200 AND v\_datum\_zaposlenja >= '1.1.2000' THEN

DBMS\_OUTPUT.PUT\_LINE

(v\_ime || ' ' || v\_prezime || ' ima plaću od ' || TO\_CHAR(v\_placa) || ' KM,  
a u tvrtki je od: ' || TO\_CHAR(v\_datum\_zaposlenja, 'dd.mm.yyyy'));

END IF;

END LOOP;

**CLOSE** djelatnik\_cursor;

**END;**

```
CREATE OR REPLACE PROCEDURE povecaj_placu  
  (p_id NUMBER, p_iznos NUMBER)  
  IS  
    v_placa NUMBER(7,2);  
    v_sifra_radmj VARCHAR2(10);  
  
  BEGIN  
    SELECT placa, sifra_radmj  
    INTO v_placa, v_sifra_radmj  
    FROM djelatnik  
    WHERE id_djelatnika = p_id;  
  
    IF NOT (v_sifra_radmj='DIR') AND v_placa + p_iznos > 3000  
    THEN  
      RAISE_APPLICATION_ERROR (-20999, 'Potrebno provjeriti iznos place!');  
    END IF;  
  
    UPDATE djelatnik  
    SET placa = placa + p_iznos  
    WHERE id_djelatnika = p_id;  
  
  END;
```

## Pohranjene procedure – *Primjer*

```
CREATE OR REPLACE PROCEDURE insert_djelatnika
(p_id_djelatnika NUMBER, p_ime VARCHAR2, p_prezime VARCHAR2,
p_spol CHAR, p_datum_zaposlenja DATE, p_placa NUMBER, p_id_sefa NUMBER,
p_sifra_opcine NUMBER, p_sifra_radmj VARCHAR2)
IS
    v_dummy VARCHAR2(10);
BEGIN
    SELECT 'x'
    INTO v_dummy
    FROM radno_mjesto
    WHERE sifra_radmj = p_sifra_radmj;
    INSERT INTO djelatnik (id_djelatnika, ime, prezime, spol, datum_zaposlenja,
        placa, id_sefa, sifra_opcine, sifra_radmj)
    VALUES (p_id_djelatnika, p_ime, p_prezime, p_spol, p_datum_zaposlenja,
        p_placa, p_id_sefa, p_sifra_opcine, p_sifra_radmj);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Ne postoji navedeno radno mjesto!');
END;
```

```
CREATE OR REPLACE FUNCTION placa_djelatnika  
  (p_id_djelatnika IN djelatnik.id_djelatnika%TYPE)  
  RETURN NUMBER  
  IS  
    v_placa djelatnik.placa%TYPE :=0;  
  BEGIN  
    SELECT placa INTO v_placa  
    FROM djelatnik  
    WHERE id_djelatnika = p_id_djelatnika;  
    RETURN v_placa;  
  END;
```

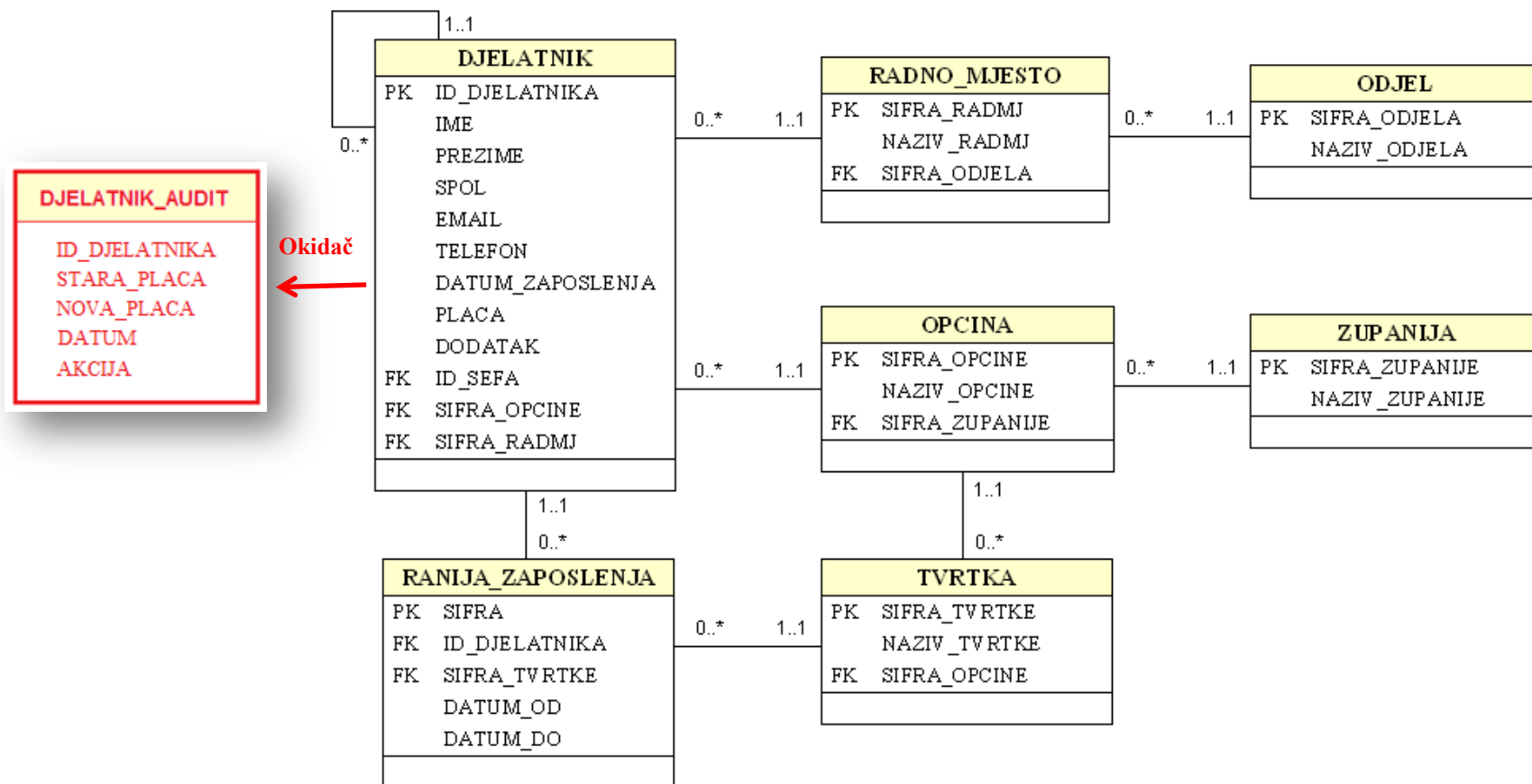
PL/SQL – Poziv funkcije:

```
DECLARE  
  id NUMBER(5) := 1;  
  placa NUMBER(7,2);  
  BEGIN  
    placa := placa_djelatnika(id);  
    DBMS_OUTPUT.PUT_LINE('Placa djelatnika ' || id || ' iznosi: ' || placa);  
  END;
```



# Testna baza

(proširena tablicom koju će puniti okidač ...)



```
CREATE OR REPLACE TRIGGER audit_djelatnika  
AFTER INSERT OR UPDATE OF placa ON djelatnik  
FOR EACH ROW  
  
BEGIN  
  
IF INSERTING THEN  
    INSERT INTO DJELATNIK_AUDIT (id_djelatnika, stara_placa, nova_placa,  
        datum, akcija)  
    VALUES (:NEW.id_djelatnika, :OLD.placa, :NEW.placa, SYSDATE, 'INSERT');  
ELSE  
    INSERT INTO DJELATNIK_AUDIT (id_djelatnika, stara_placa, nova_placa,  
        datum, akcija)  
    VALUES (:NEW.id_djelatnika, :OLD.placa, :NEW.placa, SYSDATE, 'UPDATE');  
END IF;  
  
END;
```

Obavljanje korektivne akcije preko okidača (trigera) koja bazu podataka dovodi u konzistentno stanje (referencijalni integritet!)

```
CREATE OR REPLACE TRIGGER cascade_update  
AFTER UPDATE OF sifra_odjela ON odjel  
FOR EACH ROW  
  
BEGIN  
    UPDATE radno_mjesto  
    SET radno_mjesto.sifra_odjela=:NEW.sifra_odjela  
    WHERE radno_mjesto.sifra_odjela=:OLD.sifra_odjela;  
  
END;
```

**Web:**

**<http://www2.fsr.ba/nastava/baze>**

**Pitanja, primjedbe, dogovor za konzultacije ...**

➤ E-mail: ***[goran.kraljevic@hteronet.ba](mailto:goran.kraljevic@hteronet.ba)***