ENKAPSULIRANI SQL

27.10.2013.

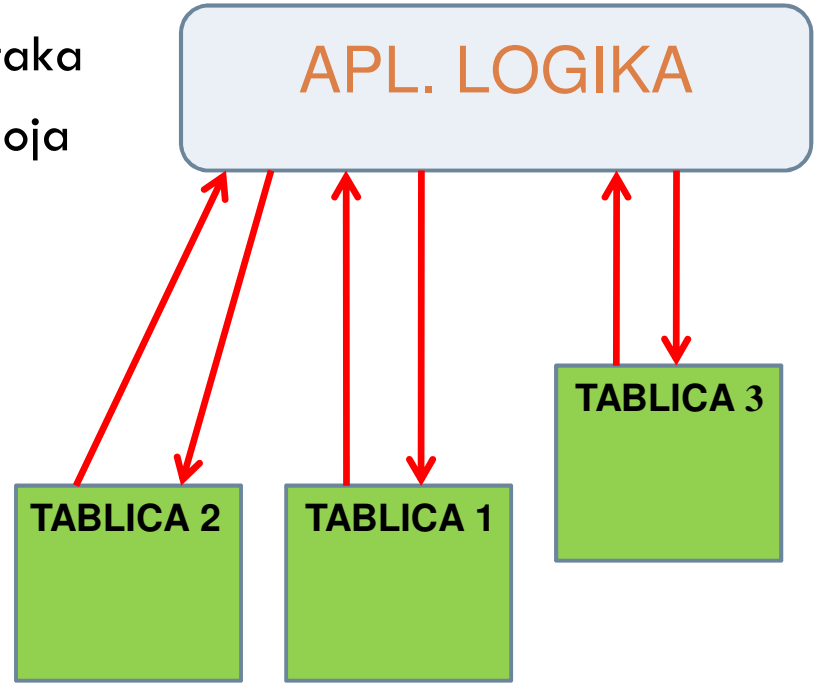
Ervin Jagatić

Pregled predavanja

- Analiza SQLa u PL/SQL-u
- Zašto je SQL loš?
- Što ponuditi kao rješenje
- Primjeri i pregledi mogućnosti upotrebe enkapsuliranog SQLa

SQL u PL/SQLu

- Backend aplikacija
  - Stalni dohvati i ažuriranje podataka
  - Izravan dohvat iz aplikativnog sloja u bazni
- SQL upiti
  - Stalna potreba za izmjenama
  - Utječu na performanse
  - Izvori runtime errora



Opasnosti kod SQLa

- Primjer
  - Napišemo koplekasn upit, iskoristimo ga u funkciji, u izvještaju, na više različitih mjesta....
  - Nakon nekog vremena shvatimo da moramo izmijeniti upit
  - SQL isto kao i HardCode

Steven Feuerstein

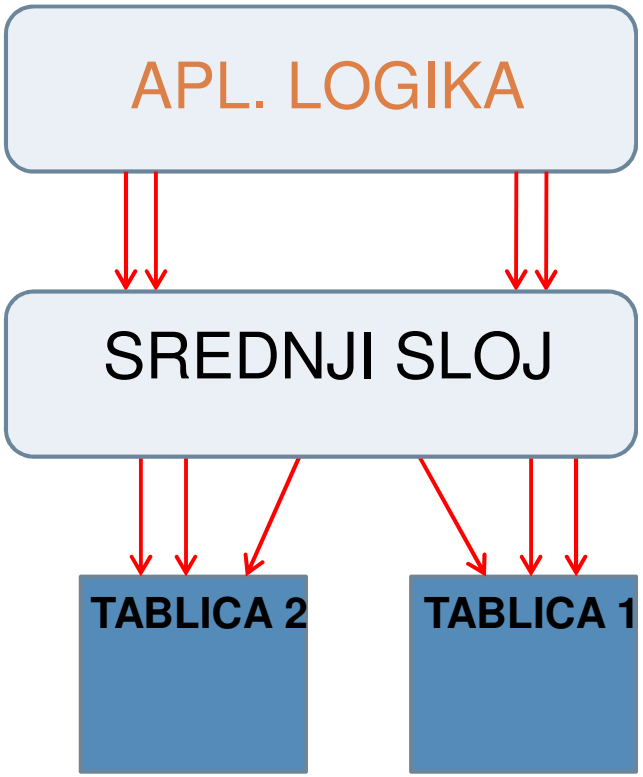
http://www.oracleplsqlprogramming.com/

Rješenje

- NE pisati SQL u aplikacijskom kodu
  - Bilo da se radi o front ili backend dijelu aplikacij
- Izbjegavati ponavljanje upita
  - Ponavljanje upita uzrokuje teško održavanje koda
- PL/SQL samo kada je potreban
  - SQL brži, lakši

Rješenje

- SQL u srednjem sloju smješten u funkcije i procedure, lako za održavati i ponvno koristiti



Primjer

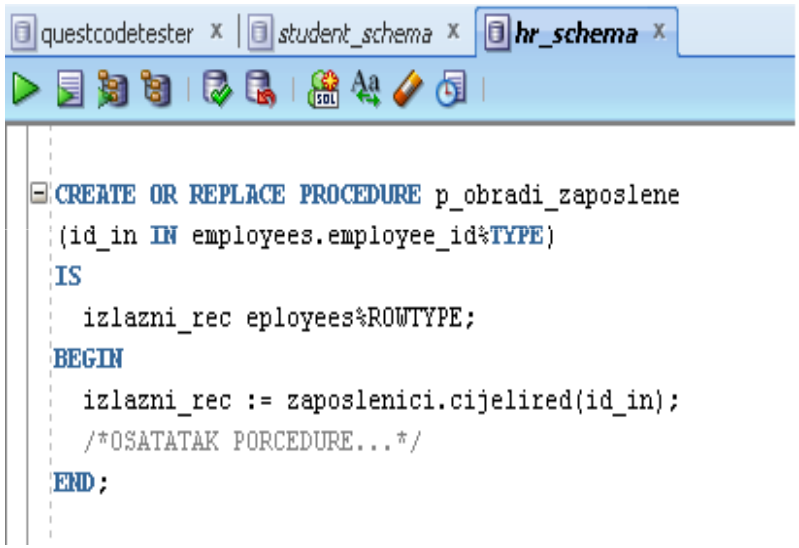
- Procedura obrade zaposlenika

questcodetester | student_schema | hr_schema

```
CREATE OR REPLACE PROCEDURE  p_obardi_zasposlene(
  id_in IN employees.employee_id%TYPE)
IS
  izlazni_rec eployees%ROWTYPE;
BEGIN
  select *
  into izlazni_rec
  from employees
  where employee_id = id_in;
  /*OSTATAK PROCEDURE.....*/
END;
```

Primjer

- Izdvojen SQL u zasebnu funkciju



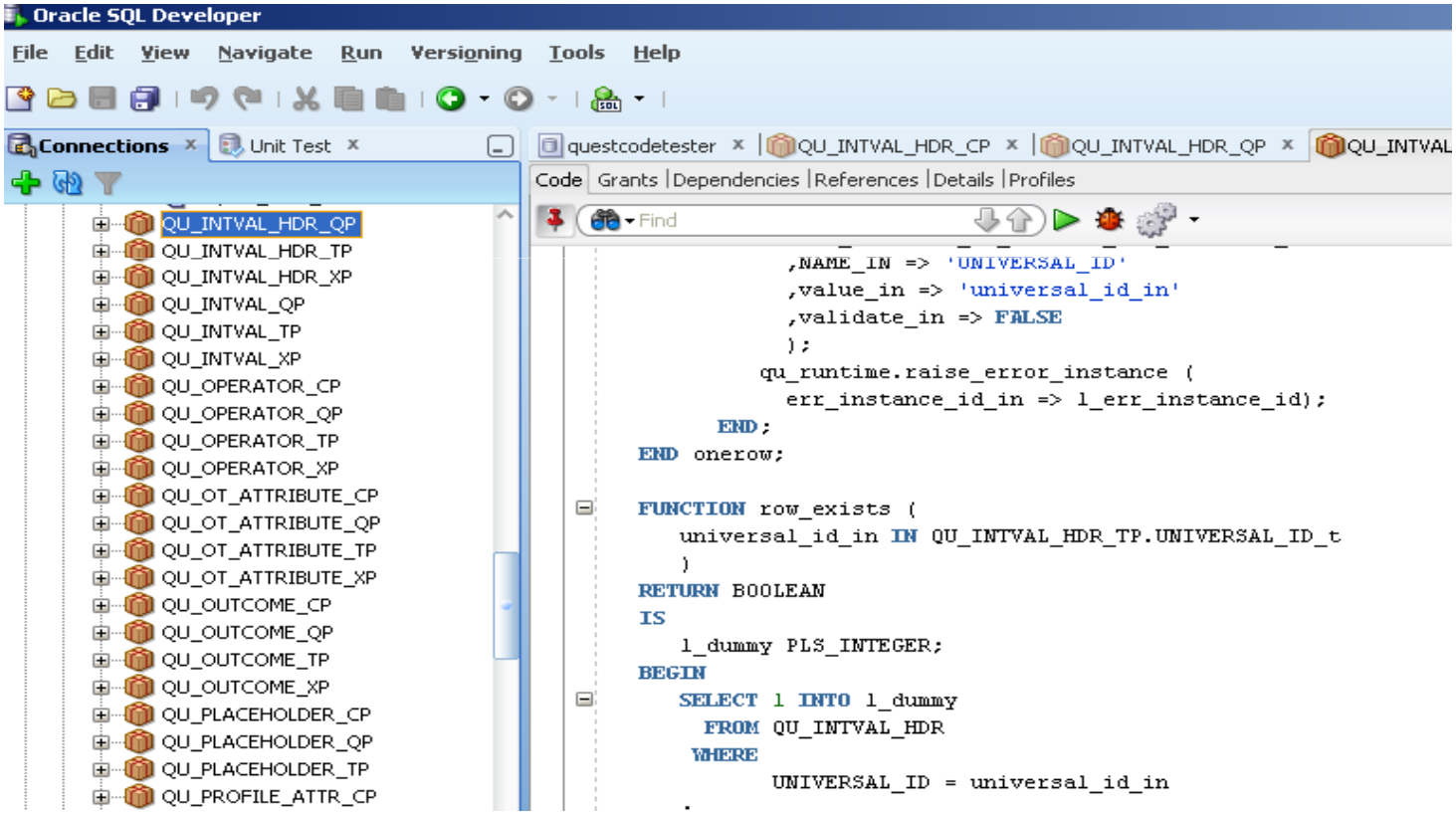

```
CREATE OR REPLACE PROCEDURE p_obradi_zaposlene
(id_in IN employees.employee_id%TYPE)
IS
  izlazni_rec eployees%ROWTYPE;
BEGIN
  izlazni_rec := zaposlenici.cijelired(id_in);
  /*OSATATAK PORCEDURE...*/
END;
```

```
CREATE OR REPLACE PACKAGE zaposlenici
IS
  FUNCTION cijelired(id_in IN employees.employee_id%type)
    RETURN employees%ROWTYPE
    --result_cache;
END;

CREATE OR REPLACE PACKAGE BODY zaposlenici
IS
  FUNCTION cijelired(id_in IN employees.employee_id%type)
    RETURN employees%ROWTYPE
    result_cache relies_on(employees)
  IS
  izlazni_rec eployees%ROWTYPE;
BEGIN
  select *
  into izlazni_rec
  from employees
  where employee_id = id_in;
  /*OSTATAK PROCEDURE.....*/
END;
```

Primjer

- Formiranje paketa



Primjer

- Upotreba enkapsuliranog SQLa kod reporta

Zaključak

- Mijenjamo implementaciju aplikacije s minimalnom intervencijom u kodu
- Poboljšana obrada run-time grešaka vezanih uz SQL
- Moguća bolja implementacija poslovnih pravila

Zaključak

- SQL se ne može izbjeći u aplikacijama na Oracle bazama
- Izbjegavati SQL izraze u aplikacijskoj logici aplikacija
- Sakriti SQL iz PL\SQL APIa