APEX Express Edition + Row level security

Mario Jozak
Oracle Certified Professional
mario.jozak@king-ict.hr

Problem iz prakse

- Izrada aplikacije koju koristi više klijenata na istoj bazi, a podaci klijenata su osjetljive prirode
- Nemogućnost korištenja VPD-a (XE)
- Osigurati development okolinu da ne postoji sigurnosni propust
- Referencijalni integritet

Rješenje

- Kreiranje dvije scheme, jedna koja će sadržavati podatke i druga koja će služiti za poslovnu logiku
- Korištenje Aplication Context-a
- Administracija APEX-a:
  - Initialization PL/SQL Code
  - Cleanup PL/SQL Code

Primjer

- Radi jednostavnosti uzet ćemo vrlo jednostavan primjer, no model se može primijeniti i na kompleksnije aplikacije, naročito kada je riječ o osjetljivim podacima

- HROUG Demo aplikacija:
  - Imamo više grupa korisnika u aplikaciji, npr. ‘DBA’,‘Developer’..
  - Imamo listu aplikacija s kojima može služiti određena grupa korisnika
  - Logirani korisnik ima pravo na aplikacije iz samo jedne grupe

Slide 4

A1	Ne Radi nego Zbog Author; 30.9.2013.
A2	Primijeniti Author; 30.9.2013.
A3	Osob ito Author; 30.9.2013.
A4	Bez Imamo? Postoji Author; 30.9.2013.
A5	Bez Imamo - Postoji Author; 30.9.2013.

Kreiranje osnovne scheme

DATA SCHEMA

- Nužna prava:
  - **CONNECT, RESOURCE**
  - **CREATE PROCEDURE**
    - context package
  - **CREATE ANY CONTEXT**
    - kreiranje context-a
  - **CREATE TABLE**
    - Tablice sa podacima
  - **CREATE VIEW**
    - View-ovi koji služe za prikaz/izmjenu/unos/brisanje podataka od strane aplikacijske scheme

Kreiranje aplikacijske scheme

APP SCHEMA

- Schema služi za pisanje poslovne logike
- Schema služi kao APEX parsing schema
- Nužna prava za ovaj model row-level sigurnosti
  - **`CONNECT, RESOURCE`**

►►►

Podaci o grupama korisnika

- Potrebno je kreirati običan šifarnik o grupama korisnika
- Taj (all-master) šifarnik će biti vezan na sve ostale tablice u bazi

```
create table user_group (
  id number(15) not null primary key,
  group_name varchar2(100) not null
)
/
```

```
SQL> insert into user_group (id, group_name) values (1, 'DBA');
1 row inserted

SQL> insert into user_group (id, group_name) values (2, 'Developer');
1 row inserted
```

Podaci o korisnicima

- Potrebno je kreirati tablicu sa korisnicima koji imaju pravo pristupa aplikaciji i u kojoj je definirana grupa na koju korisnik ima pravo

```
create table user_account (
  id number(15) not null primary key,
  username varchar2(100) not null,
  password varchar2(100) not null,
  usgr_id number(15) not null,
  constraint usracc_usgr_fk foreign key (usgr_id)
    references user_group (id),
  constraint usracc_username_ch check
    (username = upper(username) and instr(username, ' ')=0),
  constraint usracc_username_uq unique (username)
)
/
```

►►►

Podaci o korisnicima – vol2

- Demo podaci
  - Korisnik 'MARIO.VRHOVAC' može pristupiti podacima grupe 'DBA'

```
SQL> insert into user_account (id, username, password,
usgr_id) values (1, 'MARIO.VRHOVAC', 'king', 1);
1 row inserted
```

  - Korisnik 'MARIO.JOZAK' može pristupiti podacima grupe 'Developer'

```
SQL> insert into user_account (id, username, password,
usgr_id) values (2, 'MARIO.JOZAK', 'peasant', 2);
1 row inserted
```

Application Context

- Da bi omogućili automatsko filtriranje zapisa potrebno je kreirati application context
- Za kreiranje context-a nije potrebno da package (u našem primjeru ‘p#context’) unaprijed bude kreiran već ga je moguće i naknadno kreirati
- U ovom primjeru namespace context-a će biti ‘DATA’ i koristiti ćemo package p#context.

```
SQL> create context data using p#context;
Context created
```

A6 Kako bi

Author; 30.9.2013.

Context Package

- Package koji će služiti za postavljanje context-a ima dvije public procedure
  - Procedura za postavljanje context-a

```
procedure setContextForUser(
    p_username user_account.username%type
);
```

  - Procedura za brisanje context-a

```
procedure clearContext;
```

- Po potrebi, u ovom package-u mogu složiti i kompleksnije procedure za postavljanje context-a, no za ovaj primjer su potrebne dvije procedure

A7 Bez zareza

Author; 30.9.2013.

Procedura setContextForUser

- Procedura ima jedan ulazni parametar i to je korisničko ime
- Procedura treba na temelju ulaznog parametra pročitati grupu korisnika kojoj je korisnik dodijeljen i postaviti context

Procedura setContextForUser vol.2

```
procedure setContextForUser(
  p_username user_account.username%type
)
is
begin
  dbms_session.set_context(
    namespace => 'DATA',
    attribute => 'USGR_ID',
    value => p#context.getUserGroupForUser(p_username),
    username =>  v('USER'),
    client_id => sys_context ('USERENV', 'CLIENT_IDENTIFIER'));
end;
```

Procedura clearContext

- Procedura clearContext jednostavno briše podatke iz context-a

- APEX pristupa bazi tako da dijeli database session pool-a. Nužno je resetirati (obrisati) context prije nego se session dodijeli nazad u session pool, tako da bi osigurali da niti jedna informacija neće procuriti

```
procedure clearContext
 is
 begin
   dbms_session.clear_context ('DATA_CONTEXT');
 end;
```

Lista aplikacija

- Do sada su složeni svi preduvjeti za kreiranje tablica ovisnih o context-u
- Za primjer ćemo kreirati tablicu sa listom aplikacija koje korisnik može koristiti

```
create table application (
  id number(15) not null primary key,
  application_name varchar2(100) not null,
  usgr_id number(15)
    default sys_context('DATA','USGR_ID') not null,
  constraint app_usgr_fk foreign key (usgr_id)
    references user_group (id)
)
/
```

Tablica application – default value

- Ključan dio za ovaj način definiranja row-level security-a je postavljanje defaultne vrijednosti na kolonu USGR_ID

```
default sys_context('DATA','USGR_ID')
```

- Nakon što kreiramo view nad tablicom *application* u kojem ćemo 'izbaciti' kolonu usgr_id, osigurali smo da će se u kolonu samo spremati vrijednosti definirane kroz package p#context

View v#application

- Definicija view-a nad tablicom application jer prikaz svih kolona osim USGR_ID kolone
- Ključan dio je filtriranje – tako da view prikazuje samo podatke iz trenutno postavljenog context-a

```
create view v#application as
select id, application_name
from application
where usgr_id = sys_context('DATA','USGR_ID')
```

Grants

- S obzirom da se poslovna logika piše u odvojenoj schemi (u našem slučaju APP schema), potrebno je istoj dati prava nad objektima u glavnoj schemi
- INSERT/UPDATE/DELETE/SELECT prava se daju nad view-ovima, a ne tablicama
- Na taj način smo sigurni da će cijela poslovna logika koristiti pristup podacima preko view-ova, a ne direktnim pristupom preko tablica

```
SQL> grant insert, update, delete, select on
v#application to app;
Grant succeeded
```

Aplikacije – dummy data

- Kreiramo par dummy zapisa u tablici aplikacija da bi primjer funkcionirao.
- Dvije aplikacije za DBA korisnike (usgr_id=1), tri aplikacije za Developer korisnike (usgr_id=2)
- DBA korisnici: Application DBA 1, Application DBA 2
- Developer korisnici: Application Developer 1, Application Developer 2, Application Developer 3

APEX aplikacija - preduvjet

- Parsing schema APEX aplikacije mora biti schema ‘APP’.
- Da bi security funkcionirao, potrebno je dodijeliti prava schemi ‘APP’ na izvršavanje package-a **`data.p#context`**

```
SQL> grant execute on p#context to app;
Grant succeeded
```

- Također, za validaciju korisnika, potrebno je dodijeliti pravo select-a na tablicu user_account

APEX aplikacija

- Uz preduvjet da je logiranje u aplikaciju riješeno (odnosno postoji dio koda koji validira korisničko ime i lozinku iz tablice 'DATA.USER_ACCOUNT') sve što je potrebno napraviti u postavkama APEX aplikacije se nalazi pod 'Security Attributes' (Shared Components -> Security -> Security Attributes)

- U odjeljku 'Database Session' postoje dva polja:
  - Initialization PL/SQL Code (dio koda koji se izvršava odmah nakon što je APP_USER popunjen)
  - Cleanup PL/SQL Code (dio koda koji se izvršava na kraju izvršavanja stranice)

APEX –Database Session -Init

- Initialization PL/SQL Code
  - U ovo polje je potrebno upisati proceduru koja postavlja context za logiranog korisnika.
  - Znači, context će se postavljati prilikom svakog izvršavanja stranice

```
data.p#context.setContextForUser(:APP_USER);
```

  - Poziva se procedura u data schemi, koja postavlja context USGR_ID na vrijednost iz tablice za logiranog korisnika

```
dbms_session.set_context(
     namespace => 'DATA',
     attribute => 'USGR_ID',
     value => p#context.getUserGroupForUser(p_username),
     username =>  v('APP_USER'),
     client_id => sys_context ('USERENV', 'CLIENT_IDENTIFIER'));
```

APEX –Database Session -Cleanup

- Cleanup PL/SQL Code
  - U ovo polje je potrebno upisati proceduru koja će očistiti context na kraju svakog izvršavanja stranice

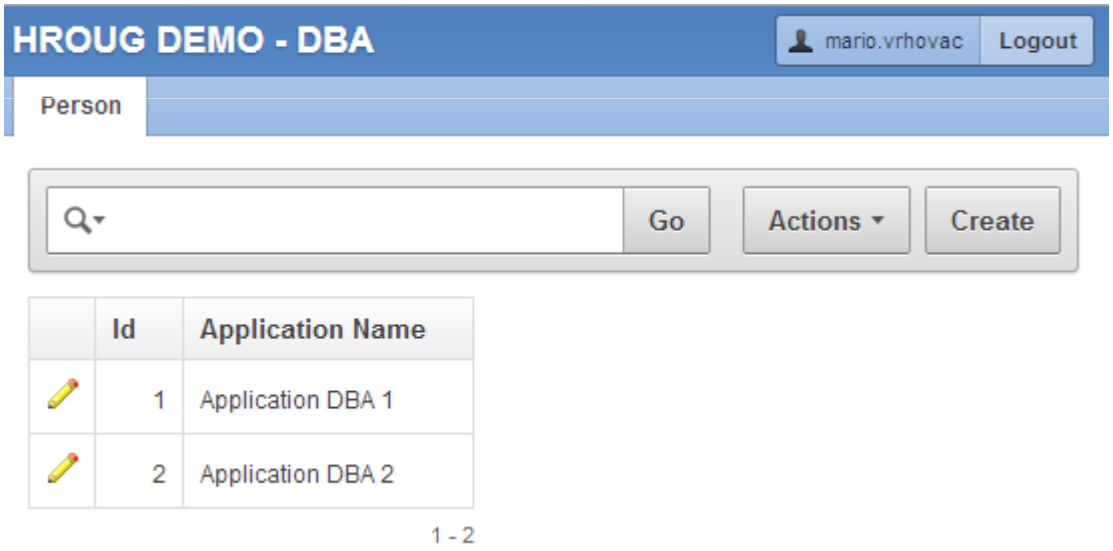
```
data.p#context.clearContext;
```

  - Poziva se procedura u data schemi koja briše context

```
dbms_session.clear_context('DATA_CONTEXT');
```

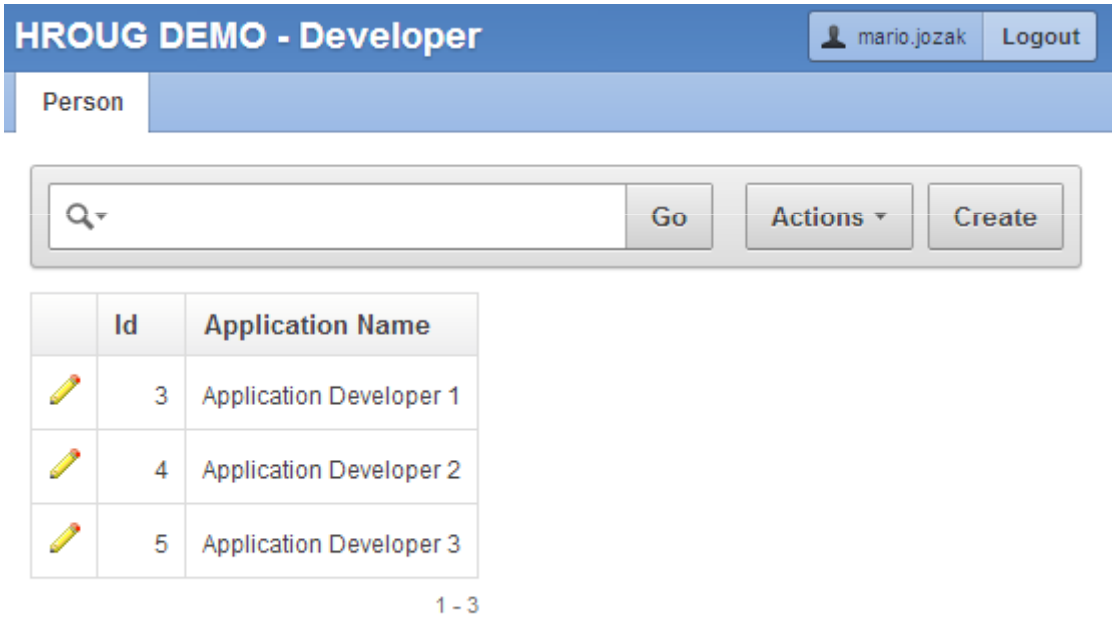
APEX - aplikacija

- Svi preduvjeti su zadovoljeni, aplikacija se može početi razvijati
- Primjer logiranog korisnika iz DBA grupe (mario.vrhovac):



APEX – aplikacija vol.2

- Primjer logiranog korisnika iz ‘Developer’ grupe (mario.jozak):



DML i referencijalni integritet

- DML
  - Unos/Izmjena/Brisanje nad tablicom ‘Application’ se radi na standardni način, no u pozadini nije tablica već view ‘V#APPLICATION’.
  - USGR_ID – iako nije ‘vidljiv’ u view-u, biti će popunjen zbog default value na toj koloni iz context-a

- Referencijalni integritet
  - Recimo da postoji tablica ‘Application_type’, koja se veže na tablicu ‘Application’.
  - Analogno tablici ‘Application’ definirati ćemo i tablicu ‘Application_type’ – default column value, updateble view..
  - Tablice se povežu kompozitnim ključem i na taj način smo osigurali da su podaci sigurno odvojeni

Razvoj poslovne logike

- Logiranjem u 'APP' schemu na bazi, sve što je potrebno je pozvati proceduru za popunjavanje context-a na temelju proizvoljnog korisnika – za onoga s čijim podacima želimo baratati:

```
SQL> conn app/xxx@kingdev_xe;
Connected to Oracle Database 11g Express Edition Release 11.2.0.2.0
Connected as app@kingdev_xe

SQL> exec data.p#context.setContextForUser('MARIO.VRHOVAC');
PL/SQL procedure successfully completed

SQL> select * from data.v#application;
                ID APPLICATION_NAME
---------------- ----------------------
                 1 Application DBA 1
                 2 Application DBA 2
```

Savjeti

- Radi lakšeg korištenja objekata u APP schemi, mogu se kreirati sinonimi za objekte iz DATA scheme
- Na taj način smo dobili isto nazivlje, pa eventualno može olakšati razvoj
- No, ovaj način zahtjeva dodatnu administraciju objekata na bazi

```
SQL> create synonym application for data.v#application;
Synonym created
```

- Sa ovom naredbom smo dobili sinonim APPLICATION u APP schemi za tablicu APPLICATION u DATA schemi (bez USGR_ID kolone)

A8 Zbog

Author; 30.9.2013.

Q&A

?

Za sva dodatna pitanja me možete kontaktirati na mario.jozak@king-ict.hr i/ili u predvorju hotela ;)