Database Programming with SQL
kurs 2017 – database design and programming with sql
students slajdovi

17-1 Controlling User Access

- Davanje dozvole (grant) ili oduzimanje dozvole (take away) za pristup db objektima je način kontrolisanja ko može izmeniti, obrisati, updejtovati, uneti, indeksirati ili referencirati db objekte

Controlling User Access

- U okruženju rada sa više korisnika, treba održati sigurnost db pristupa i korišćenja
- Sa Oracle Server db bezbednost može se raditi sledeće:
    - kontrola db pristupa
    - davanje pristupa posebnim objemtima u db
    - potvrda datih i primljenih privilegija unutar Oracle DD
    - kreiranje sinonima za db objekte

Db Security

- Db bezbednost se može podeliti na dve kategorije: sistemska bezbednost i bezbednost podataka
- System security označava pristup i korišćenje db na nivou sistema, kao što je kreiranje korisnika, usernames i lozinki, alociranje prostora diska za korisnike, dozvola sistem privilegija koje korisnici mogu koristiti kao što je kreiranje tabela, pogleda i sekvenci
- Postoji više od 100 različitih sistem privilegija
- Data security (object security) se odnosi na objekt privilegije koje označavaju pristup i korišćenje db objekata i akcije koji te korisnici mogu imati nad tim objektima
- Ove privilegije uključuju mogućnost izvršavanja DML iskaza

Privileges and Schemas

- Privilegije su prava nad izvršavanjem određenih SQL iskaza
- DBA je korisnik visokog nivoa sa mogućnostima dozvoljavanja korisnicima pristupa db i njenim objektima
- Korisnici traže sistem privilegije za dobijanje pristupa db
- Oni traže objekt privilegije za manipulaciju sadržajem objekata u db
- Korisnici mogu takođe dobiti privilegije za dozvoljavanje dodatnih privilegija za druge korisnike ili za uloge (roles), koje su imenovane grupe ili povezane privilegije
- Šeme su kolekcije objekata, poput tabela, pogleda i sekvenci; šema je vlasništvo db korisnika i ima isto ime kao i korisnik
- Na kursu naziv tvoje šeme je kombinacija tvoje country/state, school, course i student number; npr, uswa_skhs_sql01_s22

System Security

- Ovaj nivo bezbednosti pokriva pristup i korišćenje db na sistem nivou
- Postoji više od 100 različitih sistem privilegija
- Sistem privilegije kao što su mogućnost kreiranja ili otklanjanja korisnika, otklanjanje tabela ili bekap tabela najčešće ima samo DBA
- Sledeća tabela pokazuje neke od sistem privilegija koje DBA normalno neće dozvoliti drugim korisnicima

System Privilege	Operations Authorized
CREATE USER	Grantee can create other Oracle users (a privilege required for a DBA role).
DROP USER	Grantee can drop another user.
DROP ANY TABLE	Grantee can drop a table in any schema.
BACKUP ANY TABLE	Grantee can backup any table in any schema with the export utility.
SELECT ANY TABLE	Grantee can query tables, views, or snapshots in any schema.
CREATE ANY TABLE	Grantee can create tables in any schema.

System Privileges

- DBA kreira korisnika izvršavanjem CREATE USER iskaza
- Korisnik nema nijednu privilegiju u ovom momentu
- DBA može dozvoliti tražene privilegije korisniku
- Sintaksa:
  CREATE USER user
  IDNETIFIED BY password;
- Primer:

```
CREATE USER  scott
IDENTIFIED BY ur35scott;
```

- Korišćenje ALTER USER iskaza, korisnik može promeniti njihove lozinke
- Primer:

```
ALTER USER scott
IDENTIFIED BY imscott35;
```

- Ovo se ne može proveravati na APEX pošto učenici nemaju dovoljne privilegije

User System Privileges

- DBA koristi GRAN iskaz za alociranje sistem privilegija za korisnika
- Sistem privilegije određuju šta korisnik može da radi na nivou db
- Kada su korisniku dozvoljene privilegije, korisnik može odmah i da ih koristi

```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...];
```

```
GRANT  create session, create table, create sequence, create view
TO scott;
```

- Korisnik mora imati CREATE SESSION privilegiju i user id da bi bio u mogućnosti da pristupi db
- Ne može se izdati CREATE SESSION komanda u APEX-u; to se dešava automatski u pozadini

System Privilege	Operations Authorized
CREATE SESSION	Connect to the database.
CREATE TABLE	Create tables in the user's schema.
CREATE SEQUENCE	Create a sequence in the user's schema.
CREATE VIEW	Create a view in the user's schema.
CREATE PROCEDURE	Create a procedure, function, or package in the user's schema.

Object Security

- Nivo bezbednosti pokriva pristup i korišćenje db objekata i akcija koje korisnici mogi izvesti na tim objektima

Object Privileges

- Svaki objekat ima određeni set dozvoljivih privilegija

Object Privilege	Table	View	Sequence	Procedure
ALTER	X		X	
DELETE	X	X		
EXECUTE				X
INDEX	X	X		
INSERT	X	X		
REFERENCES	X			
SELECT	X	X	X	
UPDATE	X	X		

- Važno je primetiti object privileges:
  - jedine privilegije koje se odnose na sekvence su SELECT i ALTER
  - sekvenca koristi ALTER za izmenu INCREMENT, MAXVALUE, CACHE/NOCACHE ili CYCLE/NOCYCLE opcije
  - START WITH ne može biti promenjeno korišćenjem ALTER
- Može se dozvoliti UPDATE, REFERENCES i INSERT privilegija na pojedinačnim kolonama u tabeli
- Npr:

```
GRANT UPDATE (salary)
      ON employees TO steven_king
```

- SELECT privilegija može biti ograničena kreiranjem pogleda sa subsetom kolona i dozvolom SELECT privilegije samo na pogled
- Ne može se dozvoliti SELECT na pojedinačne kolone
- Privilegija dozvoljene na sinonime je pretvorena u privilegiju na osnovnu tabelu referenciranu od strane sinonima; drugim rečima, sinonim je samo nov, lakši način za korišćenje imena
- Korišćenje ovog imena za dozvolu privilegija je ista kao dozvola privilegije na samoj tabeli

PUBLIC službena reč

- Vlasnik tabele može dozvoliti pristup svim korisnicima korišćenjem PUBLIC službene reči
- Primer ispod omogućava svim korisnicima na sistemu da prave upit nad podacima iz Alice DEPARTMENTS tabele

```
GRANT select
   ON alice.departments
TO PUBLIC;
```

- Ako iskaz ne koristi puno ime objekta, Oracle Server implicitno prefiksuje ime objekta sa trenutnim imenom korisnika (ili šemom)

- Ako korisnik Scott daje upit nad DEPARTMENTS tabelom, npr, sistem radi SELECT iz SCOTT.DEPARTMENTS tabele
- Ako iskaz ne koristi puno ime objekta a trenutni korisnika nema vlasnička prava nad objektom toga imena, sistem prefiksuje ime objekta sa PUBLIC
- Npr, ako korisnik Scott daje upit nad USER_OBJECTS tabelom, a Scott nema u vlasništvu takvu tabelu, sistem selektuje iz DD pogled na način PUBLIC.USER_OBJECTS javni sinonim

Confirming Granted Privileges

- Ako pokušaš izvesti nedozvojenu operaciju, poput brisanja reda iz tabele na koju nemaš DELETE privilegije, Oracle Server ne dozvoljava izvršenje takve operacije
- Ako se dobije od Oracle servera poruka o grešci "table or view does not exist" to znači da je pokušano ili imenovanje tabele ili pogleda koji ne postoji ili pokušano izvođenje operacije na tabeli ili pogledu za koji nepostoje dovoljne privilegije

View Privileges

- Može se pristupiti DD za videti privilegije koje imaš
- Prikazana karta opisuje različite DD poglede
- Korišćenje APEX, unesi SQL Workshop, Utilities, Object Reports
- Korisničke privilegije se mogu videti u Security Reports sekciji

Data Dictionary View	Description
ROLE_SYS_PRIVS	System privileges granted to roles
ROLE_TAB_PRIVS	Table privileges granted to roles
USER_ROLE_PRIVS	Roles accessible by the user
USER_TAB_PRIVS_MADE	Object privileges granted on the user's objects
USER_TAB_PRIVS_RECD	Object privileges granted to the user
USER_COL_PRIVS_MADE	Object privileges granted on the columns of the user's objects
USER_COL_PRIVS_RECD	Object privileges granted to the user on specific columns
USER_SYS_PRIVS	Lists system privileges granted to the user

17-2 Creating and Revoking Object Privileges

Roles

- Uloga je imenovana grupa povezanih privilegija koje mogu da se dozvole korisniku
- Ovaj metod čini lakšim pozivanje (revoke) i održavanje privilegija
- Korisnik može imati pristup nekoliko uloga, a nekoliko korisnika može imati dodeljenu istu ulogu
- Uloge su tipično kreirane za db aplikacije
- Korišćenje uloga čini proces dozvola privilegija lakšim. Umesto da se dodeljuju individualne privilegije hiljadama korisnika, DBA može napraviti ulogu, dodeliti privilegije toj ulozi a zatim dozvoliti ulogu korisnicima

```
CREATE ROLE manager;
Role created.

GRANT create table, create view TO manager;
Grant succeeded.

GRANT manager TO jennifer_cho;
Grant succeeded.
```

- Za kreiranje uloge: CREATE ROLE role_name;
- Pošto je uloga kreirana, DBA može koristiti GRANT iskaz za dodelu uloge korisnicima kao i za dodelu privilegija ulozi
- Sleeći primer kreira manager role i onda dopušta menadžeru da napravi tabele i poglede
- Zatim dozoljava ulogu korisniku; a sada korisnik može kreirati tabele i poglede

```
CREATE ROLE manager;
Role created.

GRANT create table, create view TO manager;
Grant succeeded.

GRANT manager TO jennifer_cho;
Grant succeeded.
```

- Ako korisnik ima više dozvoljenih uloga, oni primaju sve privilegije dodeljene svim tim ulogama
- CREATE ROLE je sistem privilegija koji nije dostupan na kursu

Characteristics Of Roles

- Uloge su imenovane grupe povezanih privilegija i one se mogu dozvoliti korisnicima
- One pojednostavljuju proces dozvole i opoziva (revoking) privilegija

Granting Object Privileges

- Koristiti sintaksu za dozvolu objekt privilegija:

```
GRANT object_priv [(column_list)]
  ON object_name
TO {user|role|PUBLIC}
  [WITH GRANT OPTION];
```

Syntax	Defined
object_priv	is an object privilege to be granted
column_list	specifies a column from a table or view on which privileges are granted
ON object_name	is the object on which the privileges are granted
TO user|role	identifies the user or role to whom the privilege is granted
PUBLIC	grants object privileges to all users
WITH GRANT OPTION	Allows the grantee to grant the object privileges to other users and roles

Object Privileges Guidelines

- Za dozvoljavanje privilegija na objektu, objekat mora biti u tvojoj šemi ili moraš imati dozvolu privilegije korišćenja WITH GRANT OPTION

- Vlasnik objekta može dozvoliti bilo koji objekat privilegiju na objekat bilo kom drugom korisniku ili ulozi u db
- Vlasnik objekta automatski dobija sve objekat privilegije na tom objektu

GRANT Examples

- Scott King (username scott_king) has created a clients table.
- In Example 1 on the right, all users are granted permission to SELECT from Scott's clients table.
- Example 2 grants UPDATE privileges to Jennifer and to the manager role on specific columns in Scott's clients table.

```
1. GRANT SELECT
   ON clients
   TO PUBLIC;

2. GRANT UPDATE(first_name,
                last_name)
   ON clients
   TO jennifer_cho, manager;

3. SELECT  *
   FROM scott_king.clients;

4. CREATE SYNONYM clients
   FOR scott_king.clients;

5. SELECT *
   FROM clients;
```

- If Jennifer now wants to SELECT data from Scott's table, the syntax she must use is listed in Example 3.
- Alternatively, Jennifer could create a synonym for Scott's table and SELECT from the synonym.
- See the syntax in Examples 4 and 5.

- Different object privileges are available for different types of schema objects.
- A user automatically has all object privileges for schema objects contained in his schema.
- A user can grant any object privilege on any schema object that the user owns to any other user or role.

WITH GRANT OPTION

- Privilegija koja je dozvoljena korišćenjem WITH GRANT OPTION iskaza može da se preda drugom korisniku i ulogama
- Objekat privilegije dozvoljene korišćenjem WITH GRANT OPTION iskaza se opozivaju kada se privilegija opoziva
- Sledeći primer daje korisniku Scott pristup mojoj klijent tabeli sa privilegijama za upit nad tabelom i dodavanje redova u tabelu a takođe mu daje dozvolu da drugima pruži iste privilegije:

```
GRANT   SELECT, INSERT
ON      clients
TO      scott_king
WITH    GRANT OPTION;
```

PUBLIC službena reč

- Vlasnik tabele može dozvoliti pristup svim korisnicima korišćenjem PUBLIC reči

- Preimer omogućava svim korisnicima u sistemu da prave upite nad podacima iz Jasonove tabele client:

```
GRANT   SELECT
ON      jason_tsang.clients
TO      PUBLIC;
```

DELETE Object

- Ako se pokuša izvesti neautorizovana operacija kao brisanje reda iz tabele na kojoj nema DELETE privilegija, Oracle Server neće dopustiti operaciju da se izvrši
- Ako se primi Oracle Server poruka o grešci "table or view does not exists" urađeno je jedna od dve mogućnosti:
  - referencirana je tabela ili pogled koji ne postoje
  - pokušano je izvesti operaciju na tabeli ili pogledu za koje nema odgovarajućih privilegija

Revoking Object Privileges

- Može se odstraniti privilegije dozvoljene drugim korisn korišćenjem REVOKE iskaza
- Kada se koristi REVOKE iskaz, privilegije koje su specificirane su opozvane od korisnika koji su imenovani i od drugih korisnika čije su privilegije dozvoljene korišćenjem WITH GRANT OPTION iskaza
- Korišćenje sintakse opoziva objekat privilegije:

```
REVOKE {privilege [, privilege...]|ALL}
ON  object
FROM   {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```

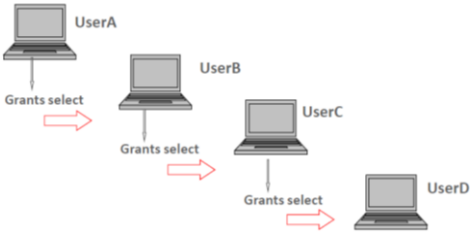
- CASCADE CONSTRAINTS je zahtevano za odstranjivanje bilo kojeg referenciranog integritet ograničenja napravljenog na objektu pomoću REFERENCES privilegija

With Grant Option

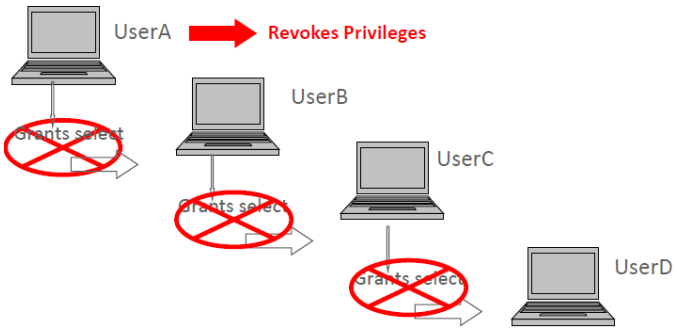
- Sledeći primer opoziva SELECT i INSERT privilegije date korisniku Scott na tabeli clients:

```
REVOKE SELECT, INSERT
ON clients
FROM scott_king;
```

- Ako je korisniku dozvoljena privilegija sa WITH GRANT OPTION iskazom, taj korisnik takođe može dozvoliti privilegiju korišćenja WITH GRANT OPTION iskaza
- To znači da dugačan lanac dozvola je moguć, ali kružne dozvole nisu dopuštene
- Ako vlasnik opozove privilegiju od korisnika koji je dozvolio privilegije drugim korisnicima, iskaz opoziva kaskadno se širi prema svim dozvoljenim privilegijama
- Npr, ako je korisnik A dao dozvolu SELECT privilegija na tabelu korisniku B uključujući WITH GRANT OPTION iskaz, korisnik B može dozvoliti korisniku C SELECT privilegije uključujući WITH GRANT OPTION iskaz takođe



- Sada, korisnik C može dozvoliti korisniku D SELECT privilegije
- Ipak, ako korisnik opoziva privilegije od korisnika B, onda te privilegije dozvoljene korisniku C i D su takođe opozvane



Private and Public Synonyms

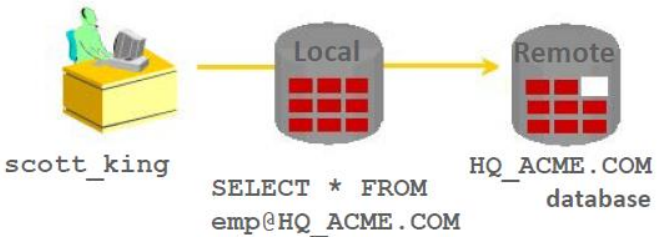
- Kao što je ranije rečeno, može se napraviti sinonim za eleminisanje potreba za kvalifikacijom imena objekata sa šemom i dati alternativno ime za tabelu, pogled, sekvencu, proceduru ili druge objekte
- Sinonimi mogu biti ili privatni (po difoltu) ili javni
- Javni sinonimi mogu biti kreirani od DBA ili db korisnika koji ima privilegije za to, ali ne može svako automatski da kreira javne sinonime
- CREATE PUBLIC SYNONYM privilegije nisu dopuštene za Oracle studente

Roles and Privileges

- Uloge i privilegije se razlikuju na nekoliko načina:
  - korisnička privilegija je pravo za izvršenje određenog tipa SQL iskaza ili pravo za pristup drugom korisničkom objektu
  - sve privilegije su definisane od Oracle
  - uloge, sa druge strane, su kreirane od korisnika (DBA) i koriste se za grupisanje privilegija ili drugih uloga
  - one su kreirane za olakšavanje upravljanja dozvolama više privilegija ili uloga korisnicima
  - privilegije dolaze sa db a uloge se prave od DBA ili korisnika određene db

Database Links

- Db link je pointer koji definiše jednosmerni komunikacioni put od jedne Oracle db do druge db
- Link pointer je zapravo definisan kao ulaz u DD tabelu
- Za pristup linku, mora se biti konektovan na lokalnu db koj asadrži DD ulaz



- Db link konekcija je jednosmerna u smislu da klijent konektovan na lokalnu db A može koristiti link smešten u db A za pristup informacijama u udaljenom db B, ali korisnici konektovani u db B ne mogu koristiti isti link za pristup podacima iz db A

- CREATE DATABASE LINK – u APEX, ne postoji konstantna konekcija sa db pa to i nije moguće ostvariti
- Ako lokalni korisnik na db B želi da pristupi podacima na db A, mora definisati link koji je smešten u DD od db B
- Db link konekcija daje lokalnim korisnicima pristup podacima na udaljenim db
- Da bi se ovakva konekcija pojavila, svaka db u distributivnom sistemu mora imati jedinstveno globalno db ime
- Globalno db ime jedinstveno identifikuje db server u distributivnom sistemu
- Velika prednost db link je to što oni omogućavaju korisnicima pristup drugim korisničkim objektima u udaljenim db tako da su oni ogrnaičeni privilegijskim setom vlasnika objekata
- Drugim rečima, lokalni korisnik može pristupiti udaljenim db bez da mora biti korisnik na udaljenoj db
- Primer pokazuje korisnika scott_king koji pristupa EMP tabeli na udaljenoj db sa globalnim imenom HQ.ACME.COM
- Tipično, DBA je odgovoran za kreiranje db link
- DD pogled USER_DB_LINKS sadrži informacije o linkovima kojima korisnik ima pristup
- Kada je jednom db link kreiran, može se pisati SQL iskaz nad podacima na udaljenom sajtu
- Ako je sinonim postavljen, može se napisati SQL iskaz korišćenjem sinonima
- Npr:

```
CREATE PUBLIC SYNONYM HQ_EMP
 FOR emp@HQ.ACME.COM;
```

- Then write a SQL statement that uses the synonym:

```
SELECT *
FROM HQ_EMP;
```

- You cannot grant privileges on remote objects.

17-3 Regular Expression

- Ponekad je potrebno pronaći ili zameniti određeni komad teksta u koloni, tekst string ili dokument
- Do sada je prikazano kako se radi sa jednostavnim upoređivanjem šablona korišćenjem LIKE i wildcards
- Nekad je potrebno tražiti kompleksne tekst stringove kao ekstrakovati sav URL iz dela teksta
- Regularni izrazi (RE) su metode opisivanja jednostavnih i kompleksnih šablona za traženje i manipulaciju
- Oraklova implementacija RE je ekstenzija POSIX (Portable Operating System for UNIX) i potpuno je kompatabilan sa POSIX standardima, koje kontroliše IEEE

Regular Expressions

- Korišćenje RE je bazirano na upotrebi meta karaktera
  - meta karakteri su specijalni znakovi koji imaju posebno značenje, poput wildcard karaktera, ponavljajućeg karaktera, non-matching karaktera ili opsega karaktera

- može se koristiti nekoliko predefinisanih meta karaktera simbola u odgovarajućim šablonima

META Characters

Symbol	Description
. (dot)	Matches any character in the supported character set, except NULL
?	Matches zero or one occurrence
*	Matches zero or more occurrences
+	Matches one or more occurrences
()	Grouping expression, treated as a single sub-expression
\	Escape character
|	Alternation operator for specifying alternative matches
^/$	Matches the start-of-line/end-of-line
[ ]	Bracket expression for a matching list matching any one of the expressions represented in the list

Regular Expression Examples

- Jednostavan RE je veoma sličan wildcard pretragama
- Primer: koristiti dot operator za traženje slova a posle kojeg je bilo koji karakter posle koga je slovo c
- Kao RE, to bi izgledalo kao: a.c
- Isti izraz kao standardni SQL wildcard search bi bio: WHERE column LIKE 'a_c'
- Koji od sledećih primera bi odgovarao a.c ? Odgovori u crvenom su dobri:
- 'ABC', 'abc', 'aqx', 'axc', 'aBc', 'abC' 'Amc' 'amrc'
- Drugi primeri nisu ili zato što im je karakter na pogrešnoj poziciji ili u pogrešnoj veličini

Regular Expression Functions

- Oracle omogućava set SQL funkcija koje se mogu koristiti za pretragu i manipulaciju stringova korišćenjem RE
- Ove funkcije se mogu koristiti na bilo kojem tipu podataka koji sadrži karakter podatak kao što su CHAR, CLOB, VARCHAR2
- RE mora biti zatvoren pod apostrofima

Name	Description
REGEXP_LIKE	Similar to the LIKE operator, but performs regular expression matching instead of simple pattern matching
REGEXP_REPLACE	Searches for a regular expression pattern and replaces it with a replacement string
REGEXP_INSTR	Searches for a given string for a regular expression pattern and returns the position where the match is found
REGEXP_SUBSTR	Searches for a regular expression pattern within a given string and returns the matched substring
REGEXP_COUNT	Returns the number of times a pattern appears in a string. You specify the string and the pattern. You can also specify the start position and matching options (for example, c for case sensitivity).

Regular expression Function Examples

- Pretpostaviti da treba izlistati sve zaposlene sa imenom Stephen ili Steven

```
SELECT first_name, last_name
FROM employees
WHERE REGEXP_LIKE(first_name, '^Ste(v|ph)en$');
```

- Sa RE samo će se koristiti REGEXP_LIKE funkcija i search string '^Ste(v|ph)en$'
  - ^ specificira start stringa koji se traži
  - zatim idu tri slova Ste
  - sa (počinje podizraz a u njemu je slovo v, znak| koje specificira OR, slova ph i) koje zatvara podizraz
  - zatim dva slova en
  - $ specificira krajk stringa koji se traži
- Primer:

```
SELECT first_name, last_name
FROM employees
WHERE REGEXP_LIKE(first_name, '^Ste(v|ph)en$');
```

- Result:

FIRST_NAME	LAST_NAME
Steven	King

- RE REPLACE funkcija će zameniti jedan string šablon sa drugim
- Ovaj primer traži "H" a posle njega bilo koji samoglasnik (vowel), i zamenjuje ih sa dva "*":

```
SELECT last_name, REGEXP_REPLACE(last_name, '^H(a|e|i|o|u)', '**')
  AS "Name changed"
FROM employees;
```

LAST_NAME	Name changed
Abel	Abel
Davies	Davies
...	...
Hartstein	**rtstein
Higgins	**ggins
...	...

- RE COUNT funkcija vraća broj puta koliko se šablon pojavljuje u stringu
- Primer pretražuje za subizraz "ab":

```
SELECT country_name, REGEXP_COUNT(country_name, '(ab)') AS "Count of 'ab'"
FROM wf_countries
WHERE REGEXP_COUNT(country_name, '(ab)')>0;
```

COUNTRY_NAME	Count of 'ab'
Republic of Zimbabwe	1
Arab Republic of Egypt	1
Great Socialist Peoples Libyan Arab Jamahiriya	1
Kingdom of Saudi Arabia	1
Syrian Arab Republic	1
Gabonese Republic	1
United Arab Emirates	1

Regular Expressions in Check Constraints

- RE mogu takođe da se koriste kao deo koda aplikacije za osiguranje da samo validan podatak je smešten u db
- Moguće je uključiti poziv RE funciji u, npr CHECK ograničenje

Regular Expressions in Check Constraints

- Tako da ako je potrebno osigurati da ne postoji nijedna email adresa bez '@' u tabeli u db, može se samo dodati sledeća check ograničenje:

```
ALTER TABLE employees
ADD CONSTRAINT email_addr_chk
CHECK(REGEXP_LIKE(email,'@'));
```

- Ovo će osigurati da email obavezno ima '@' znak
- Korišćenje RE može se proveriti format email adrese mnogo rigoroznije za proveru validnosti
- Validan email adresa će imati jedan ili više kartaktera nego jedno @, posle kojeg ide jedno ili više karaktera nego . (dot) praćeno sa jednim ili više karaktera

```
CREATE TABLE my_contacts
(first_name VARCHAR2(15),
 last_name VARCHAR2(15),
 email VARCHAR2(30) CHECK(REGEXP_LIKE(email, '.+@.+\..+')));
```

- Syntax definitions:
  - .+ means one or more characters
  - @ an @ symbol
  - \. a . (a dot) (here the backslash is an escape character)