Rozdział 16. Uprawnienia, role, synonimy – zadania

Zadania należy wykonywać parami, jeden użytkownik działa jako użytkownik A zaś drugi jako użytkownik B. W kilku ćwiczeniach konieczna jest obecność trzeciego użytkownika C.

1. Użytkownik A próbuje odczytać dane z relacji ***pracownicy*** należącej do użytkownika B i *vice versa*.

   ```
   ORA-00942: table or view does not exist
   ```

2. Użytkownik B nadaje użytkownikowi A prawo odczytu danych z własnej relacji ***pracownicy***.

3. Użytkownik A ponownie próbuje odczytać dane z relacji ***pracownicy*** należącej do użytkownika B.

4. Użytkownik A nadaje użytkownikowi B prawo modyfikowania atrybutów *placa_pod* i *placa_dod* we własnej relacji ***pracownicy***.

5. Użytkownik B modyfikuje zawartość relacji ***pracownicy*** należącej do użytkownika A. Następnie tworzy prywatny synonim dla relacji ***pracownicy*** należącej do A i modyfikuje kolejne krotki tej relacji za pomocą synonimu. Po zakończeniu modyfikacji użytkownik B zatwierdza swoje zmiany poleceniem COMMIT.

6. Użytkownik B próbuje odczytać dokonane przez siebie zmiany w relacji ***pracownicy*** użytkownika A.

7. Użytkownicy A i B oglądają informacje ze słownika bazy danych dotyczące przyznanych uprawnień obiektowych:

   ```
   select owner, table_name, grantee, grantor, privilege
   from   user_tab_privs;

   select table_name, grantee, grantor, privilege
   from   user_tab_privs_made;

   select owner, table_name, grantor, privilege
   from   user_tab_privs_recd;

   select owner, table_name, column_name, grantee, grantor, privilege
   from   user_col_privs;

   select table_name, column_name, grantee, grantor, privilege
   from   user_col_privs_made;

   select owner, table_name, column_name, grantor, privilege
   ```

```
from   user_col_privs_recd;
```

8. Użytkownik A odbiera użytkownikowi B prawo modyfikacji własnej relacji pracownicy. B następnie próbuje modyfikować (bezpośrednio i za pomocą synonimu) relację ***pracownicy*** należącą do A.

9. Użytkownicy tworzą role i nadają tym rolom prawo odczytu i modyfikowania danych we własnych relacjach ***pracownicy***. Rola użytkownika A powinna być chroniona hasłem, rola użytkownika B jest rolą bez hasła. Nazwy ról powinny zostać skonstruowane przez dodanie do słowa *ROLA_* numeru indeksu studenta, np. *ROLA_12345* dla użytkownika *INF12345*.

10. Użytkownik A nadaje stworzoną przez siebie rolę użytkownikowi B. B próbuje odczytać zawartość relacji ***pracownicy*** należącej do A.

11. Użytkownik B włącza rolę przyznaną mu przez użytkownika A. B próbuje odczytać zawartość relacji ***pracownicy*** należącej do A. B przegląda informacje ze słownika bazy danych dotyczącą uprawnień związanych z rolami.

```
select granted_role, admin_option from user_role_privs
where  username = 'B';

select role, owner, table_name, column_name, privilege
from   role_tab_privs;
```

12. Użytkownik A odbiera użytkownikowi B rolę. Użytkownik B próbuje odczytać informacje z relacji ***pracownicy*** należącej do użytkownika A.

13. Użytkownik B odłącza się od bazy danych, przyłącza ponownie i próbuje dokonać odczytu danych z relacji ***pracownicy*** użytkownika A.

14. Użytkownik A próbuje dokonać modyfikacji danych relacji ***pracownicy*** użytkownika B.

15. Użytkownik B nadaje użytkownikowi A rolę, jaką utworzył w p. 9. Użytkownik A ponownie próbuje dokonać modyfikacji danych relacji ***pracownicy*** użytkownika B.

16. Użytkownik A odłącza się od bazy danych, przyłącza ponownie i znowu próbuje dokonać modyfikacji danych relacji ***pracownicy*** użytkownika B.

17. Użytkownik B odbiera swojej roli prawo modyfikowania własnej relacji ***pracownicy***. Użytkownik A ponownie próbuje dokonać modyfikacji danych relacji ***pracownicy*** należącej do użytkownika B.

18. Obaj użytkownicy usuwają utworzone przez siebie role.

19. Użytkownik A nadaje użytkownikowi B prawo odczytu własnej relacji ***pracownicy*** wraz z prawem dalszego przyznawania tego uprawnienia. Użytkownik B przekazuje to prawo dalej użytkownikowi C. Użytkownik C próbuje odczytać dane z relacji ***pracownicy*** należącej do użytkownika A.

20. Wszyscy użytkownicy (A, B i C) oglądają zawartość słownika bazy danych dotyczącą nadanych i otrzymanych uprawnień obiektowych.

21. Użytkownik A próbuje odebrać uprawnienia do swojej relacji ***pracownicy*** użytkownikowi C. Następnie próbuje odebrać te same uprawnienia użytkownikowi B. Użytkownicy raz jeszcze oglądają słownik bazy danych.

22. Użytkownik A tworzy perspektywę ***prac20*** udostępniającą nazwiska i płace pracowników zespołu 20. Następnie przenosi całość uprawnień do odczytu i modyfikacji z relacji ***pracownicy*** na utworzoną perspektywę ***prac20***. Użytkownik B modyfikuje relację ***pracownicy*** użytkownika A za pomocą udostępnionej mu perspektywy.

23. Użytkownik A tworzy w swoim schemacie funkcję PL/SQL o nazwie ***funLiczEtaty***, która policzy liczbę rekordów relacji ***etaty*** i zwróci ją jako wynik. Następnie nadaje prawo wykonywania tej funkcji użytkownikowi B.

24. Użytkownik B wykonuje funkcję ***funLiczEtaty***, a następnie próbuje zweryfikować poprawność jej wyniku licząc za pomocą zapytania SQL rekordy w relacji ***etaty*** użytkownika A.

25. Użytkownik A ponownie tworzy funkcję ***funLiczEtaty***, jednak teraz funkcja ma działać z prawami użytkownika bieżącego (wykonującego) – z klauzulą `AUTHID CURRENT_USER`. Następnie ponownie nadaje prawo wykonywania tej funkcji użytkownikowi B.

26. Użytkownik B ponownie wykonuje funkcję ***funLiczEtaty***. Czy otrzymany wynik różni się od wyniku wykonania funkcji z punktu 24.?

27. Użytkownik A dodaje do swojej relacji ***etaty*** nowy rekord, opisujący etat o nazwie WYKŁADOWCA i pensji od 1000 do 2000 zł. Po dodaniu rekordu A zatwierdza operację poleceniem COMMIT.

28. Użytkownik B ponownie wykonuje funkcję ***funLiczEtaty***. Dlaczego otrzymany wynik nie różni się wyniku wykonania funkcji z punktu 26.?

29. Użytkownik B tworzy relację ***test*** o schemacie: *id* number(2), *tekst* varchar2(20) i dodaje do niej dwa rekordy: (1,'pierwszy'), (2, 'drugi'). Następnie tworzy procedurę ***procPokazTest***, której zadaniem jest wypisanie na konsoli zawartości kolumny tekst ze wszystkich rekordów relacji ***test***. Procedura ma działać z uprawnieniami bieżącego użytkownika (klauzula `AUTHID CURRENT_USER`). Następnie B nadaje użytkownikowi A prawo wykonywania procedury ***procPokazTest***.

30. Użytkownik A próbuje uruchomić procedurę ***procPokazTest*** użytkownika B. Dlaczego operacja kończy się niepowodzeniem? Co powinien zrobić użytkownik B, aby A mógł wykonać procedurę ***procPokazTest***?

31. Utwórz relację ***info_dla_znajomych*** o poniższym schemacie:

```
NAZWA VARCHAR2(20) NOT NULL
INFO VARCHAR2(200) NOT NULL
```

Wpisz do relacji kilka krotek. Jako wartości atrybutu *nazwa* podaj identyfikatory innych użytkowników w bazie danych. Utwórz perspektywę ***info4u*** i nadaj do niej odpowiednie prawa w ten sposób, aby każdy użytkownik bazy danych mógł odczytać z perspektywy ***info4u*** informacje przeznaczone tylko i wyłącznie dla siebie.