|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ | |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА» | |
|  | |
|  | |
| Лабораторная работа №21 | |
|  | |
|  | Выполнил:  Стерпу Е.К., гр. 6411  Проверил:  Додонов М.В. | |
|  | |
| Самара 2018 | |

**Цель:** обучение работе с механизмом «тщательный контроль доступа» (*Fine Grained Access Control – FGAC)*, являющимся важным компонентом системы безопасности СУБД *Oracle*.

**Порядок выполнения работы**

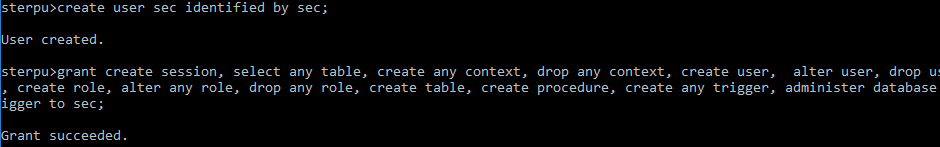
1. Соединяемся с базой данных СУБД Oracle как SYS с параметром AS SYSDBA.

sterpu> connect system as sysdba

1. Создаем пользователя SEC с паролем SEC.

Simon> create user sec identified by sec;

1. Разрешаем пользователю SEC создавать сессию с опцией администратора, выбирать данные из любой таблицы, создавать любой контекст, удалять любой контекст, создавать пользователей, изменять пользователей, удалять пользователей, создавать роли, изменять любые роли, удалять любые роли, создавать таблицы, создавать процедуры, создавать любые триггеры, создавать триггеры на базу данных.

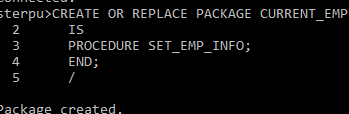
Разрешаем пользователю SEC использовать пакет DBMS\_SESSION.

sterpu> grant execute on dbms\_session to sec;

1. Соединяемся с базой данных СУБД Oracle как SEC.



1. Создаем пакет CURRENT\_EMP, который содержит в себе описание процедуры SET\_EMP\_INFO.

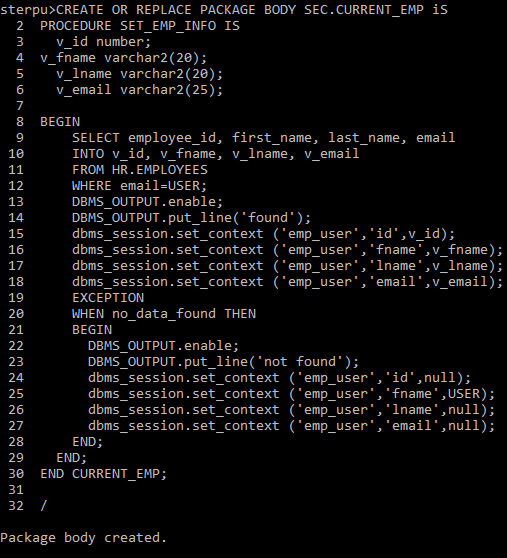


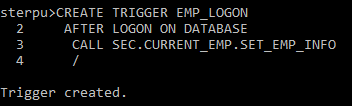
1. Создаем контекст EMP\_USER используя пакет CURRENT\_EMP.

sterpu> CREATE CONTEXT EMP\_USER

2 USING CURRENT\_EMP;

Context created.

1. Создаем содержимое пакета CURRENT\_EMP и описываем в нем логику работы процедуры SET\_EMP\_INFO. В процедуре SET\_EMP\_INFO формируется запрос к таблице HR.EMPLOYEES, который выбирает строку, содержащую EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME из таблицы HR.EMPLOYEES, для которой EMAIL == SESSION\_USER и заносит данные из этой строки в контекст EMP\_USER следующим образом: ID =EMPLOYEE\_ID, NAME = FIRST\_NAME||’ ‘||LAST\_NAME, EMAIL = SESSION\_USER. В случае если искомая строка в таблице не найдена, процедура должна возвращать значение NULL. 
2. Создаем триггер EMP\_LOGON, который после соединения с базой данных вызывает процедуру SET\_EMP\_INFO, которая содержится в пакете CURRENT\_EMP.



1. Создаем пользователя SKING с паролем SKING.

sterpu> create user sking identified by sking;

1. Разрешаем пользователю SKING создавать сессию.

sterpu> grant create session to sking;

Grant succeeded.

1. Соединяемся с базой данных Oracle как SKING.

sterpu> connect sking

1. Включаем вывод сообщений для пакета DBMS\_OUTPUT с помощью директивы SERVEROUTPUT.

Simon> set serveroutput on

1. Выводим на экран ID, NAME и EMAIL, которые содержатся в контексте EMP\_USER.

SELECT

sys\_context('emp\_user','id') id,

sys\_context('emp\_user','fname') first\_name,

sys\_context('emp\_user','lname') last\_name,

sys\_context('emp\_user','email') email

FROM dual

ID

--------------------------------------------------------------------------------

100

FIRST\_NAME

--------------------------------------------------------------------------------

STEVEN

LAST\_NAME

--------------------------------------------------------------------------------

KING

EMAIL

--------------------------------------------------------------------------------

SKING

1. Соединяемся с базой данных СУБД Oracle как SYSTEM с параметром SYSDBA и разрешаем пользователю SEC работать с пакетом DBMS\_RLS.

sterpu> connect system as sysdba

sterpy> grant execute on dbms\_rls to sec;

18. Соединяемся с базой данных СУБД Oracle как SEC.



19.Создаем или, если он уже существует, заменяем пакет HR\_POLICY\_PKG, который содержит в себе описание функции LIMIT\_EMP\_EMP, которая принимает параметры, необходимые для использования данной функции в процедуре ADD\_POLICY, пакета DBMS\_RLS:

* 1. OBJECT\_SCHEMA IN VARCHAR2
  2. OBJECT\_NAME IN VARCHAR2

sterpu> CREATE OR REPLACE PACKAGE HR\_POLICY\_PKG

2 IS

3 FUNCTION LIMIT\_EMP\_EMP(object\_schema in varchar2, object\_name in varchar2) return varchar2;

4 END;

5 /

1. Создаем содержимое пакета HR\_POLICY\_PKG и описываем в нем логику работы функции LIMIT\_EMP\_EMP, которую мы определили выше. Функция возвращает данные, типа VARCHAR2 , а именно, строку следующего вида: ‘EMPLOYEE\_ID=’, а дальше подставляется значение переменной ID из контекста EMP\_USER.

sterpu> CREATE OR REPLACE PACKAGE BODY HR\_POLICY\_PKG IS

2 FUNCTION LIMIT\_EMP\_EMP(object\_schema in varchar2, object\_name in varchar2)

3 RETURN varchar2 IS

4 str varchar2(25);

5 BEGIN

6 SELECT sys\_context('EMP\_USER', 'ID') INTO str FROM dual;

7 RETURN 'EMPLOYEE\_ID=' || str;

8 END;

9 END;

10 /

1. Проверяем корректность созданной нами функции, для этого, вызываем ее из пакета HR\_POLICY\_PKG и передаем ей в качестве параметров 'A' и ‘B’ соответственно. Если функция написана правильно, то на экране мы увидим незамысловатую надпись приблизительно такого вида: 'EMPLOYEE\_ID=132'.

sterpu> SET SERVEROUTPUT ON;

sterpu> DECLARE

2 res varchar2(255);

3 BEGIN

4 res:=HR\_POLICY\_PKG.LIMIT\_EMP\_EMP ('A', 'B');

5 DBMS\_OUTPUT.PUT\_LINE(res);

6 END;

7 /

EMPLOYEE\_ID=

PL/SQL procedure successfully completed.

1. Соединяемся с базой данных СУБД Oracle как SEC.

sterpu> conn sec/sec

1. С помощью процедуры ADD\_POLICY создать правило для пользователя SEC, которое будет вызывать функцию LIMIT\_EMP\_EMP, которая находится у пользователя HR каждый раз, когда пользователь SEC будет обращаться к таблице HR.EMLOYEES с помощью оператора SELECT.

sterpu> EXECUTE DBMS\_RLS.ADD\_POLICY('HR', 'EMPLOYEES', 'HR\_EMP\_POL', 'SEC', 'HR\_POLICY\_PKG.LIMIT\_EMP\_EMP', 'SELECT', FALSE, TRUE, TRUE);

1. Соединяемся с СУБД Oracle как HR.

sterpu> alter user hr account unlock;

User altered.

sterpu> alter user hr identified by hr;

User altered.

sterpu> conn hr/hr

Connected.

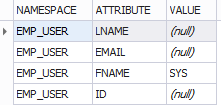
1. Разрешаем пользователю SKING осуществлять выборку данных из таблицы EMPLOYEES пользователя HR.

sterpu> grant select on hr.employees to sking;

Grant succeeded.

1. С помощью безымянного блока для всех активных контекстов вывести через пробел их NAMESPACE, ATTRIBUTE и VALUE значения. В случае, если активных контекстов нет – вывести следующее сообщение «No context active».

SELECT \* FROM V$CONTEXT



1. Осуществляем выборку всех данных из таблицы HR.EMPLOYEES. Тут необходимо заметить, что, несмотря на то, что никакого условия в запросе задано не было (фраза WHERE) опущена, СУБД вернула только запись, в которой EMPLOYEE\_ID равно номеру пользователя SKING.

sterpu>SELECT \* FROM HR.EMPLOYEES ;

EMPLOYEE\_ID

--------------------------------------------------------------------------------

100

FIRST\_NAME

--------------------------------------------------------------------------------

STEVEN

LAST\_NAME

--------------------------------------------------------------------------------

KING

EMAIL

--------------------------------------------------------------------------------

SKING

PHONE\_NUMBER

--------------------------------------------------------------------------------

515.123.4567

HIRE\_DATE

--------------------------------------------------------------------------------

17.06.2003 0:00:00

JOB\_ID

--------------------------------------------------------------------------------

AD\_PRES

SALARY

--------------------------------------------------------------------------------

24000

COMISSIOM\_PCT

--------------------------------------------------------------------------------

MANGER\_ID

--------------------------------------------------------------------------------

DEPARTMENT\_ID

--------------------------------------------------------------------------------

90

1. Соединяемся с базой данных СУБД Oracle как SEC.

sterpu> conn sec/sec

1. Удаляем созданное правило с помощью процедуры DROP\_POLICY.

sterpu> exec dbms\_rls.drop\_policy ('hr', 'employees', 'hr\_emp\_pol');