**Oracle Lab 2: User management**

***Assoc. Prof. Dr. Nguyen Thai Nghe***

Objective:

* Create/alter/drop user
* Grant permission to users
* Create role (group)
* Grant role to user

1. **Create/alter/drop users**

Syntax for creating users:

CREATE USER username

IDENTIFIED BY password

[ DEFAULT TABLESPACE tablespace ]

[ TEMPORARY TABLESPACE tablespace ]

[ QUOTA {**integer** [K | M ] | UNLIMITED } ON tablespace

[ QUOTA {integer [K | M ] | UNLIMITED } ON tablespace ] ...] ;

* ALTER USER username IDENTIFIED BY new\_password
* ALTER USER username

[ DEFAULT TABLESPACE tablespace]

[ TEMPORARY TABLESPACE tablespace]

{[ QUOTA {integer [K | M] | UNLIMITED } ON tablespace

[ QUOTA {integer [K | M] | UNLIMITED } ON tablespace ] […] };

* ALTER USER username ACCOUNT {LOCK | UNLOCK};
* ALTER USER username PASSWORD EXPIRE;
* DROP USER username

1. **User Privileges**

There are 2 kinds of privileges: System and Object privileges

***System Privileges***

| **Kinds** | **Privileges** |
| --- | --- |
| **TABLE** | **CREATE TABLE**  **CREATE ANY TABLE**  **ALTER ANY TABLE**  **DROP ANY TABLE**  **SELECT ANY TABLE**  **UPDATE ANY TABLE**  **DELETE ANY TABLE** |
| **SESSION** | **CREATE SESSION**  **ALTER SESSION**  **RESTRICTED SESSION** |
| **TABLESPACE** | **CREATE TABLESPACE**  **ALTER TABLESPACE**  **DROP TABLESPACE**  **UNLIMITED TABLESPACE** |

Syntax:

GRANT {system\_priv|role} [, {system\_priv|role} ]...

TO {user|role|PUBLIC}[, {user|role|PUBLIC} ]...

[WITH ADMIN OPTION]

REVOKE {system\_priv|role}

[, {system\_priv|role} ]...

FROM {user|role|PUBLIC}

[, {user|role|PUBLIC} ]...

***Object Privileges***

| **Privileges** | **Table** | **View** | **Procedure** |
| --- | --- | --- | --- |
| **ALTER** | X |  |  |
| **DELETE** | X | X |  |
| **EXECUTE** |  |  | X |
| **INSERT** | X | X |  |
| **SELECT** | X | X |  |
| **UPDATE** | X | X |  |

Syntax

GRANT { object\_priv [(column\_list)][, object\_priv [(column\_list)] ]...

|ALL [PRIVILEGES]}

ON [schema.]object

TO {user|role|PUBLIC}[, {user|role|PUBLIC} ]...

[WITH GRANT OPTION] ;

REVOKE { object\_priv [, object\_priv ]... | ALL [PRIVILEGES] }

ON [schema.]object

FROM {user|role|PUBLIC} [, {user|role|PUBLIC} ]...

[CASCADE CONSTRAINTS] ;

1. **Role (Group) management**

Role acts as a group. All of the users having the same privileges should be added to a role.

**Syntax**

CREATE ROLE role\_name

ALTER ROLE role\_name {NOT IDENTIFIED | IDENTIFIED

{BY password | EXTERNALLY }} ;

DROP ROLE role\_name;

**Adding user to role**

GRANT role\_name [, role\_name ]...

TO {user|role|PUBLIC} [, {user|role|PUBLIC} ]...

[WITH ADMIN OPTION] ;

REVOKE role\_name [, role\_name ]...

FROM {user|role|PUBLIC}[, {user|role|PUBLIC} ]... ;

1. **Individual Exercises. Save all of the commands to ONE FILE: *Lab2-StudentID.sql***

Login to the system using *sys* account

* Create user Test1

CREATE USER Test1 IDENTIFIED BY Test1;

* Login to the system using Test1 (getting error)



* Grant the Create Session to user Test1

GRANT CREATE SESSION TO Test1;

* Changing his password

ALTER USER Test1 IDENTIFIED BY pTest1;

* Grant permission to the user Test1.

GRANT CREATE TABLE, CREATE USER, CREATE SESSION TO Test1 **WITH ADMIN OPTION**;

* Login to the system using Test1, the try this command

CREATE TABLE Students(

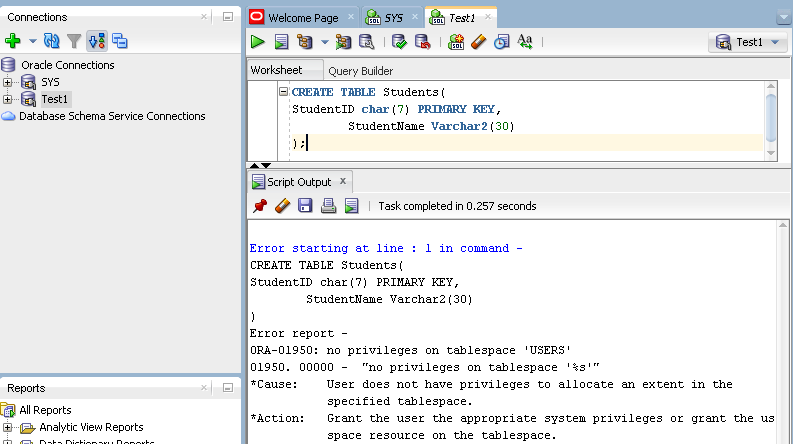
StudentID char(7) PRIMARY KEY,

StudentName Varchar2(30)

);

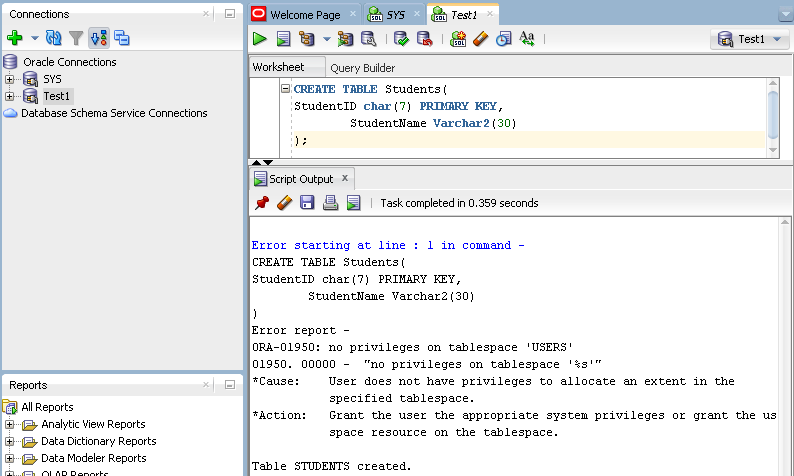
* Can he do that? Give your comments. If not, grant the quota to him then re-run Create table

**Answer: He can’t do that because user ‘Test1’ does not have privileges on tablespace ‘USERS’**



ALTER USER Test1 default tablespace users quota 2M on users;



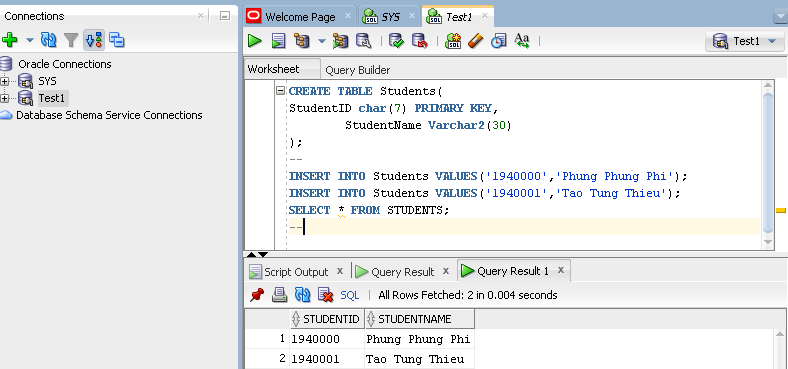


* Insert data to table

INSERT INTO Students VALUES('1940000','Phung Phung Phi');

INSERT INTO Students VALUES('1940001','Tao Tung Thieu');

SELECT \* FROM STUDENTS;



* Why the Test1 can insert data to the Students table? (he has not been granted INSERT permission)

**Answer: Because the default tablespace of user ‘Test1’ is ‘users’ and it automatically has all ‘Object Privileges’ to its own tables.**

* Check WITH ADMIN OPTION the user Test1. Login by Test1 and create another user: Test2.

CREATE USER Test2 IDENTIFIED BY Test2

DEFAULT TABLESPACE USERS QUOTA 2M ON USERS;

* Grant the CREATE SESSION to Test2

GRANT CREATE SESSION TO Test2;

* Grant permission to Test2.

GRANT CREATE TABLE TO Test2 **WITH ADMIN OPTION**;

* Login to the system using Test2 then create a table

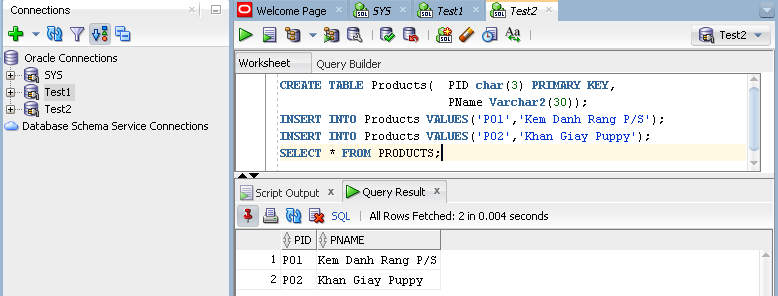
CREATE TABLE Products( PID char(3) PRIMARY KEY,

PName Varchar2(30));

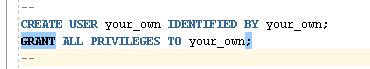
INSERT INTO Products VALUES('P01','Kem Danh Rang P/S');

INSERT INTO Products VALUES('P02','Khan Giay Puppy');

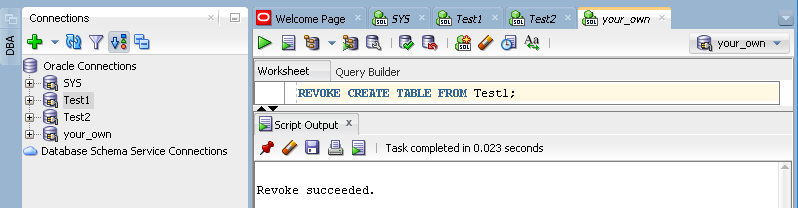
SELECT \* FROM PRODUCTS;



* Login by your\_own account to revoke the CREATE TABLE of Test1.



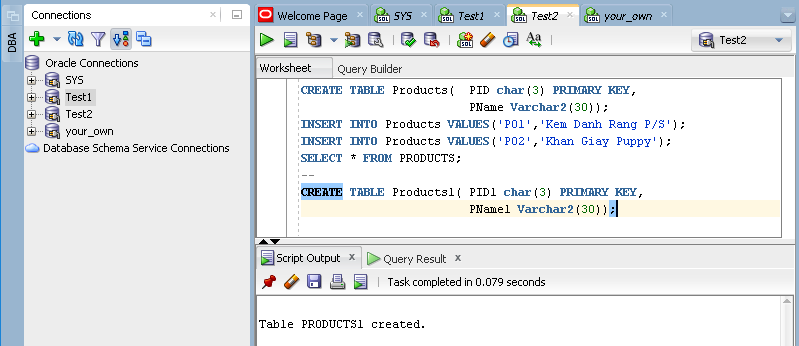
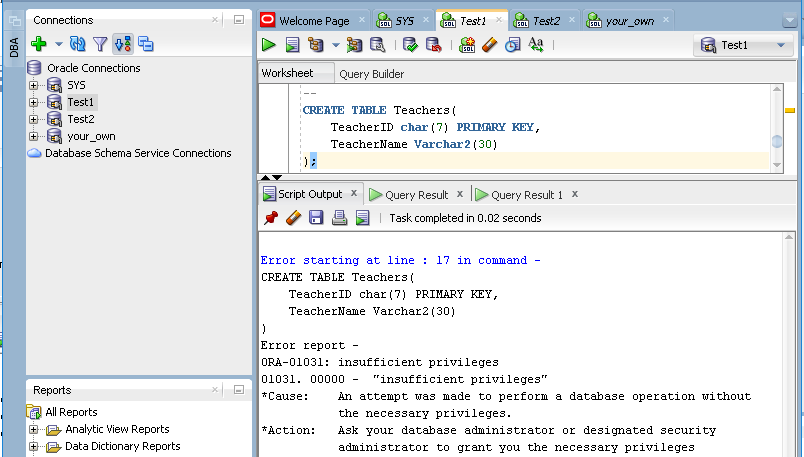
REVOKE CREATE TABLE FROM Test1;



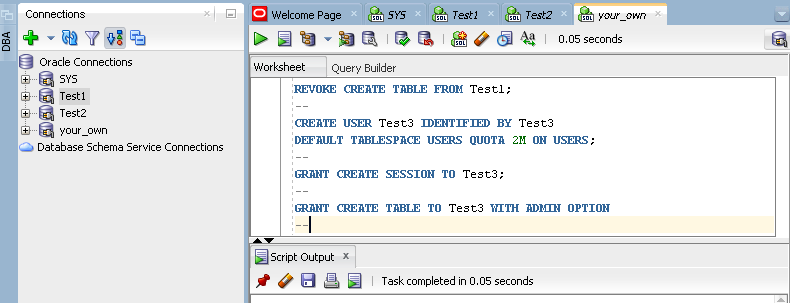
* Re-login by Test1 and Test2 to create another table. Give your comments.

**Answer:**

* **Because the user ‘your\_own’ has all privileges, he revoked the ‘create table’ privilege from user ‘Test1’.**
* **User ‘Test1’ is no longer be allowed to create tables, nothing changed from user ‘Test2’**



* Create user Test3 the same as Test2.



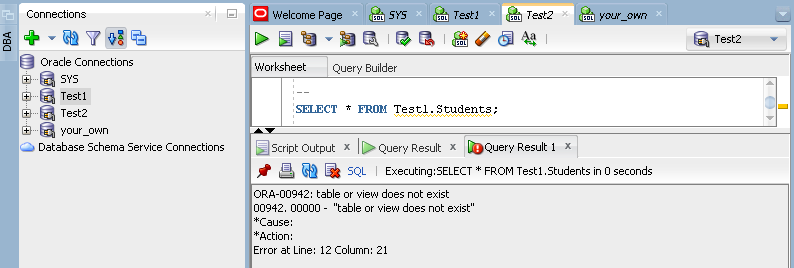
**Object privileges**

* Login to the system using Test2, then running this command

SELECT \* FROM Test1.Students;

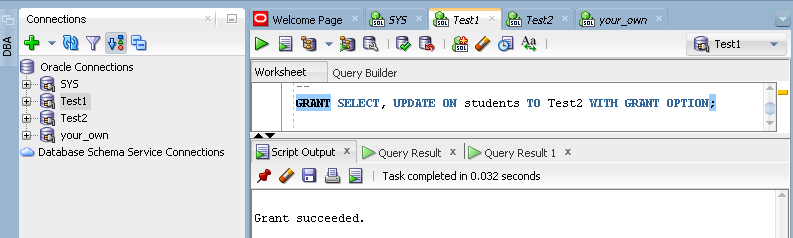
* Any errors? Why?

**Answer: User Test2 doesn’t have any privileges to table ‘Students’ of user ‘Test1’**



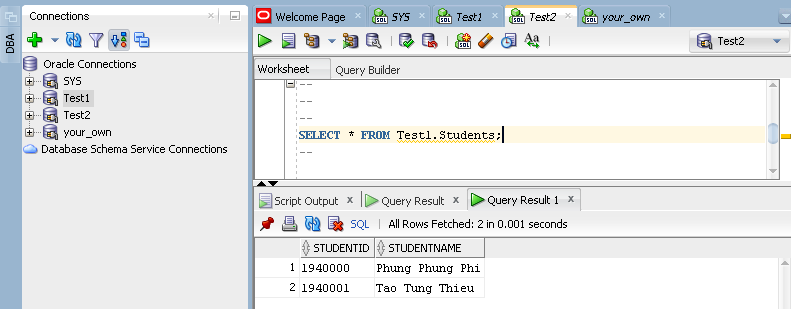
* Login by Test1 then grant SELECT, UPDATE to user Test2.

GRANT SELECT, UPDATE ON students TO Test2 **WITH GRANT OPTION**;



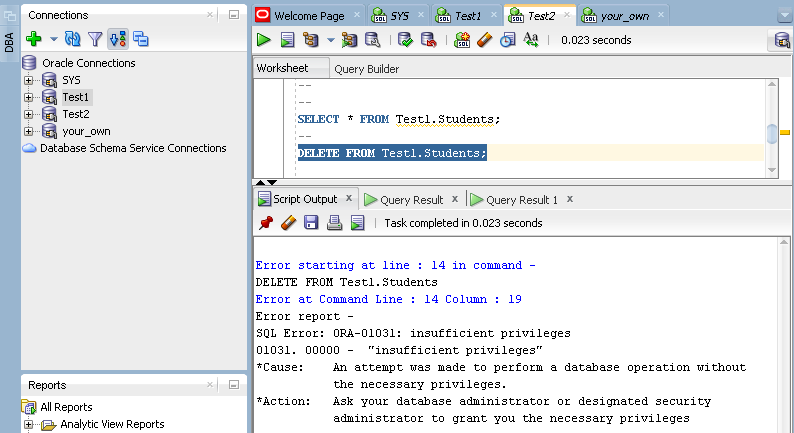
* Re-Login to the system using Test2, then running this command.

SELECT \* FROM Test1.Students;



* + Try this command

DELETE FROM Test1.Students;



Any errors? Why?

**Answer: Test2 doesn’t have the privilege ‘delete’ to table ‘Students’ of user ‘Test1’**

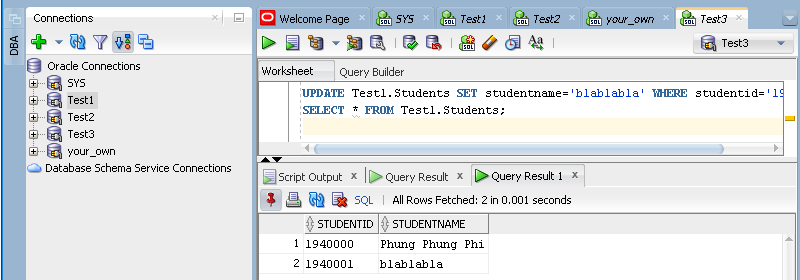
* Login by user Test2, grant SELECT, UPDATE to user Test3 on Students table

GRANT SELECT, UPDATE ON Test1.students TO Test3;

* Login by Test3 then running this command

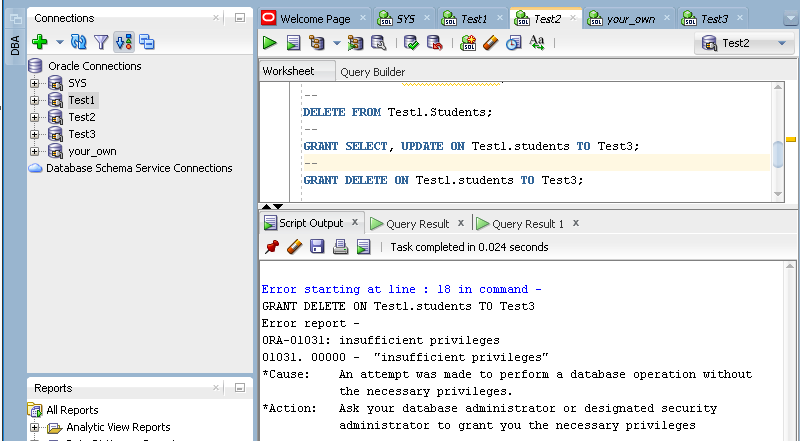
UPDATE Test1.Students SET sname=’bla bla bla’ WHERE SID='1940001'

SELECT \* FROM Test1.Students;



* Login by user Test2 then running this

GRANT DELETE ON Test1.students TO Test3;



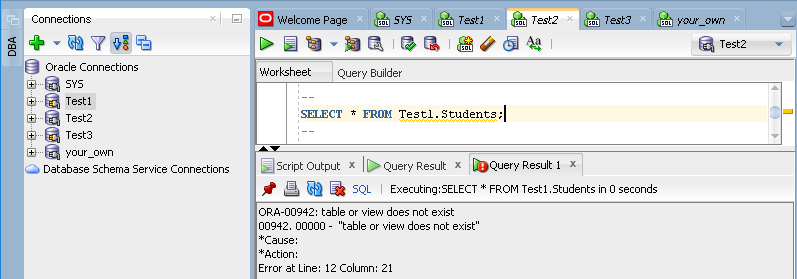
Any comments?

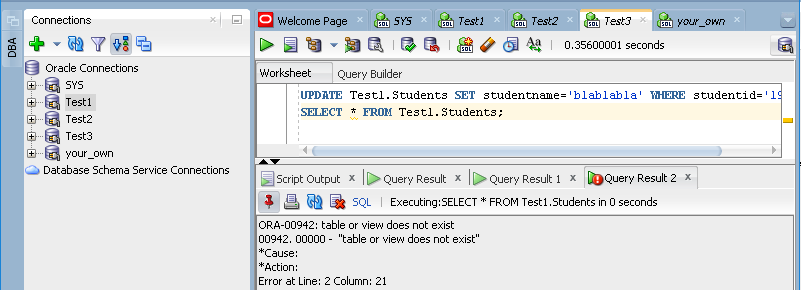
**Answer: User ‘Test2’ doesn’t have ‘delete’ privilege to the table ‘students’ of ‘Test1’. Although he already has ‘admin option’, he still cannot grant the ‘delete’ privilege on table ‘students’ to user ‘Test3’**

* Login by user Test1, revoke SELECT, UPDATE from user Test2

REVOKE SELECT, UPDATE ON Students FROM Test2;

* Running the above SQL commands on Students using Test2 and Test3, give your comments.



****

**Answer: Their privileges have been revoked.**

**Role management**

Suppose that our class has 60 students with the same permissions for connecting to Oracle, create table, and create procedure.

* Login to the system by SYS, then creating user1 and user2

CREATE USER user1 IDENTIFIED BY user1

DEFAULT TABLESPACE USERS QUOTA 1M ON USERS;

CREATE USER user2 IDENTIFIED BY user2

DEFAULT TABLESPACE USERS QUOTA 1M ON USERS;

* Create a role namely TTOracle

CREATE ROLE TTOracle;

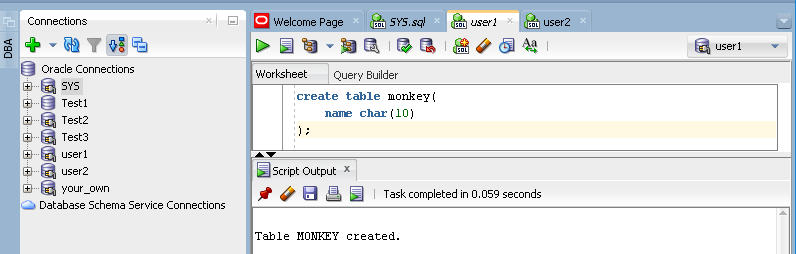
* Grant permissions to TTOracle role

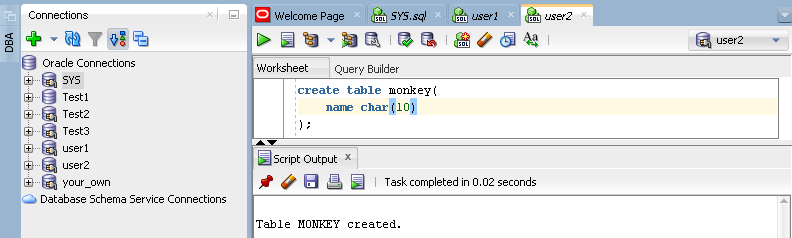
GRANT CREATE SESSION, CREATE TABLE, CREATE PROCEDURE, CREATE TRIGGER TO TTOracle;

* Adding user1 and user2 to role

GRANT TTOracle TO user1,user2;

* Login to the system using user1 or user2 to check the create table permission





* Removing user1 out of the role

REVOKE TTOracle FROM user1;

