

UNIVERSITY OF TECHNOLOGY, JAMAICA

Lab Manual

FACULTY: Engineering & Computing (FENC)
SCHOOL/DEPT: School of Computing & Information Technology
COURSE OF STUDY: Bachelor of Science in Computing
YEAR: Three (3)
MODULE TITLE: **Database Administration**

Introduction

This week you will be introduced to database **users, privileges, roles, and profiles**. We will be using two database environments: SQL Server 2008 R2, Oracle 11G. SQL Server was introduced in Advanced Databases; therefore the focus will be on Oracle this semester.

Students should store all SQL Queries in a txt file and store it on their G Drive.

Objectives:

Manage the User – Create, Alter, Drop a User

Granting Privileges – Revoke and Grant

Lab Activities:

Please Note

Each Student should create a lab report folder on your G Drive. Week by week you will write a lab report of the activities you have done in the lab. At the end of the semester you will present the lab report to your lab tutor. Weekly activities account for 10% of your grade. **All Lab Manuals are due by email before the following lab class and will be marked in the lab class.**

Lab Work should be done in groups of two. This allows students to try the various techniques on a live user. Students will observe the environment and makes notes in the Lab Report.

Managing Oracle Users

Each Oracle database has a list of valid database users. To access a database, a user must run a database application and connect to the database instance using a valid user name defined in the database. This section explains how to manage users for a database, and contains the following topics:

- Creating Users
- Altering Users
- Dropping Users

Creating Users

Note: A DBA or security administrator is usually the only one to have the privilege to create users

Example 1 - Create a User and Grant the Create Session System Privilege

Syntax:

```
CREATE USER <username> IDENTIFIED BY <password> [account lock] [password expire];
```

Live Example:

```
CREATE USER student IDENTIFIED BY pass1234;  
GRANT create session TO student;  
GRANT connect to student  
CONNECT student/pass1234
```

Note: A newly created user cannot connect to the database until granted the CREATE SESSION and CONNECT system privilege.

EXERCISE 1 – CREATE A USER

1. Create a user “student_yourid” with your ID Number appended at the end.

Observation: Using admin user, give the SQL statement used to complete the task. Find the system table for the user just created. State the table and the fields you observed.

2. Connect using the new user you have created.

Observation: Using the new user, find the system table for the user you just created. Can you see it? Give reasons for your observation.

3. Create an account for your lab partner **test1_yourid** that is locked.

Observation: Using admin user, give the SQL statement used to complete the task.

4. Allow your lab partner to attempt a log on.

Observation: Were they able to log on? What did you observe? **Do not unlock this account.**

5. Create an account for your lab partner **test2_yourid** with password expire option.

Observation: Using admin user, give the SQL statement used to complete the task.

6. Allow your lab Partner to log on using the expired account.

Observation: What did you observe?

Altering Users

Note: You have the option of changing a user's status by using alter.

Example 2 – Altering a user

Syntax:

```
ALTER USER <username> IDENTIFIED BY <password> [account un/lock] [password expire];
```

Live Example:

```
ALTER USER student IDENTIFIED BY newpass4567 account lock;
```

Note: You can lock or unlock an account.

EXERCISE 2 – ALTERING USERS

1. Alter the password for **test2_yourid** and allow your lab partner to try again

Observation: Provide the SQL statement and what you observed?

2. Alter the **test1_yourid**. Allow your lab partner to log on.

Observation: Were they able to log on? Do all that is required for them to log on and change their password. Provide the SQL statement and what you observed.

Deleting Users

Note: You have the option of deleting a user. Be aware that all tables under that user will be deleted

Example 3 – Deleting a user

Syntax:

```
DROP USER <username>;
```

Live Example:

```
DROP USER student;
```

EXERCISE 3 – DELETING USERS

1. Drop test1_yourid user.

Observation: What did you observe in the system tables for this user? Is the user gone? Use the keyword commit and then check again. Do you see any difference in the system table ?

Granting Privileges

Note: A user will not be able to access another user's schema or area unless they are granted the privilege to do so. As the DBA you will have specify that users are allowed to create objects and what objects they can create.

Example 1 – GRANT A PRIVILEGE

Syntax:

```
GRANT <ACTION> <OBJECT TYPE> to <USER>;
```

Live Example:

1. `grant create table to test_user_01;`
2. `Grant select on dummy_table to test_user_01;`

EXERCISE 4 – GRANTING PRIVILEGES - Write the appropriate observations for each task.

1. Try creating any table of your choice as student_yourid.
Observation: Provide the SQL statement and what you observed? If there was a problem creating the table how did you fix it?
2. Trying inserting 3 records.
Observation: Provide the SQL statement and what you observed? If you had an issue inserting, **do not try and fix it.**
3. Log as labuser and issue the command **ALTER USER student_yourid QUOTA UNLIMITED ON USERS.** Log on as student_yourid and try to insert the records again.
Observation: Provide the SQL statement and what you observed compared to task 4.2?
4. Allow your lab partner to try and select from your schema.
Example `Select * from test_user1.dummy_table`
Allow your lab partner to try and do the following with your table:
 - Update
 - Delete
 - Insert
 - Select
Observation: Provide the SQL statement and what you observed? Were you able to execute the statements successfully? If not do not grant any privileges as yet to fix the situation.

5. Grant your lab partner the right privileges so they will have access to your table FOR EACH STATEMENT and have them try again.

Observation: Provide the SQL statements and what you observed?

6. Create a dummy user along with a dummy table. Try to drop the user that was just created.

Observation: Were you able to drop the user? How would you fix the situation to ensure that this is done? [Hint: An Additional command is needed with drop user]