
DBMS LAB 02 MATERIAL

Prepared by:
Mohammad Anas Jawad
Lecturer, IUT CSE



Department of Computer Science and Engineering
Islamic University of Technology
January 22, 2020

Contents

1	Creating a new user	3
2	SQL Statements	4
2.1	The CREATE TABLE statement	4
2.2	The SELECT Statement	4
2.3	The INSERT INTO Statement	5
2.4	The DROP TABLE statement	6
3	Working with primary keys	7
4	Tips and Tricks	8
4.1	Checking current user	8
4.2	Disconnect/Log out	8
4.3	Locked account solution	8
4.4	Saving your work	9
4.5	Clearing the console	9

1 CREATING A NEW USER

To create a new user, we need to follow a sequence of operations.

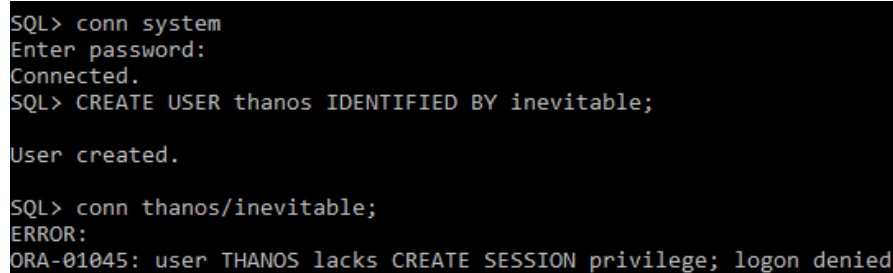
1. Login as system.
2. Create a new user with password.
3. Grant sufficient permissions/privileges.

Three major types of privileges among some others are:

1. CREATE SESSION (allows the user to login).
2. RESOURCE (allows the user to manipulate tables).
3. DBA (Database admin permission).

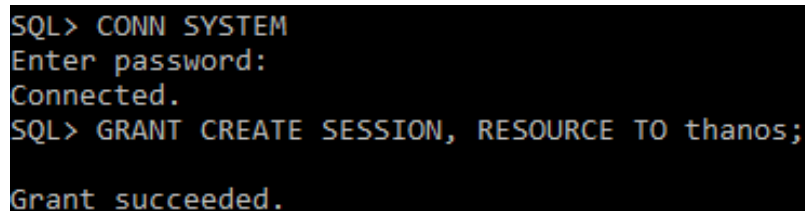
The statement to create a new user and grant sufficient privileges:

```
CREATE USER username IDENTIFIED BY password;  
GRANT privilege1, privilege2... TO username;
```



```
SQL> conn system  
Enter password:  
Connected.  
SQL> CREATE USER thanos IDENTIFIED BY inevitable;  
  
User created.  
  
SQL> conn thanos/inevitable;  
ERROR:  
ORA-01045: user THANOS lacks CREATE SESSION privilege; logon denied
```

Figure 1: Creating a new user



```
SQL> CONN SYSTEM  
Enter password:  
Connected.  
SQL> GRANT CREATE SESSION, RESOURCE TO thanos;  
  
Grant succeeded.
```

Figure 2: Granting it sufficient permissions

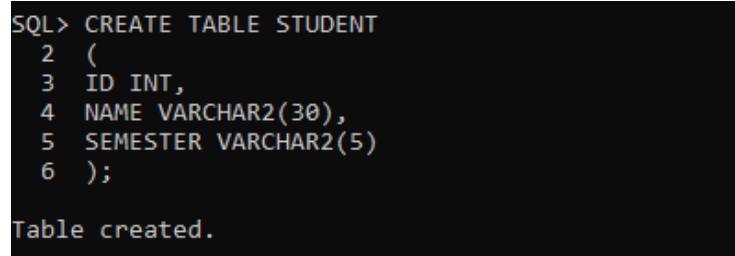
2 SQL STATEMENTS

2.1 The CREATE TABLE statement

The way you can create tables in an oracle database is by using the CREATE TABLE statement. The structure of the statement is as follows:

```
CREATE TABLE table_name
(
    column_1 data_type,
    column_2 data_type,
    .
    .
    .
);
```

Example:

A screenshot of a terminal window with a black background and white text. It shows a SQL command to create a table named 'STUDENT' with three columns: 'ID' of type 'INT', 'NAME' of type 'VARCHAR2(30)', and 'SEMESTER' of type 'VARCHAR2(5)'. The command is entered in six lines, and the response 'Table created.' is shown on the seventh line.

```
SQL> CREATE TABLE STUDENT
2  (
3  ID INT,
4  NAME VARCHAR2(30),
5  SEMESTER VARCHAR2(5)
6  );
Table created.
```

Figure 3: Creating a table

2.2 The SELECT Statement

The purpose of the SELECT statement is to display the rows and columns of a table. The very basic structure of the SELECT statement is as follows:

```
SELECT * FROM table_name;
```

"table_name" being the name of your table.

Writing this statement will result in showing the all the columns and all the rows currently

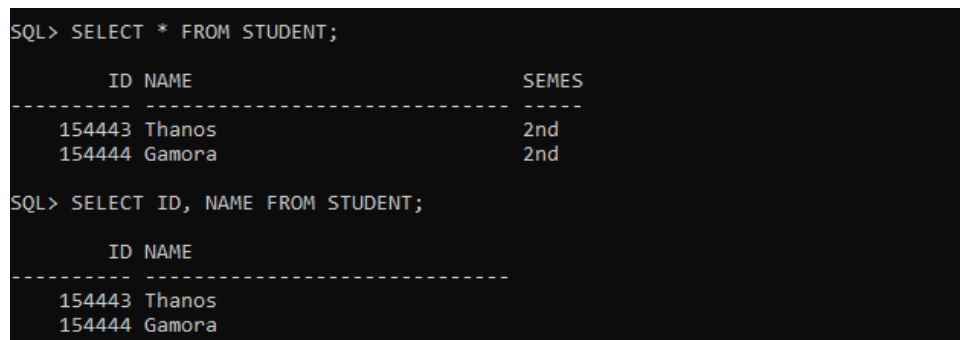
present in the database. We can also modify this statement to show us specific rows, in a sorted order etc. However, we won't be doing those in this lab.

To specify only certain columns to be displayed, write the following statement:

```
SELECT column_1, column_2 FROM table_name;
```

"column_1" and "column_2" being the columns you want to display.

Example:



```
SQL> SELECT * FROM STUDENT;
```

ID	NAME	SEMES
154443	Thanos	2nd
154444	Gamora	2nd

```
SQL> SELECT ID, NAME FROM STUDENT;
```

ID	NAME
154443	Thanos
154444	Gamora

Figure 4: The SELECT statement

There are further variations of the SELECT statement that use WHERE, ORDER BY etc. We will see these variations in the subsequent labs.

2.3 The INSERT INTO Statement

This statement is used to enter data into a table. There are two major ways of writing the INSERT INTO statement.

```
INSERT INTO table_name
```

```
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table_name (column1, column2, column3, ...)
```

```
VALUES (value1, value2, value3, ...);
```

Example:

```
SQL> INSERT INTO STUDENT VALUES (154443, 'Anas Jawad', '2nd');  
1 row created.
```

Figure 5: INSERT statement variation 1

```
SQL> INSERT INTO STUDENT (ID,NAME,SEMESTER) VALUES (154444, 'Tamzid', '2nd');  
1 row created.  
SQL> INSERT INTO STUDENT (NAME) VALUES ('Tamzid');  
1 row created.
```

Figure 6: INSERT statement variation 2

2.4 The DROP TABLE statement

This statement is used to delete a table (including all its entries) from your database.

```
DROP TABLE table_name;
```

Warning: Using this statement will remove all the entries from your table, so be careful while using it.

```
SQL> DROP TABLE STUDENT;  
Table dropped.  
SQL> SELECT * FROM STUDENT;  
SELECT * FROM STUDENT  
      *  
ERROR at line 1:  
ORA-00942: table or view does not exist
```

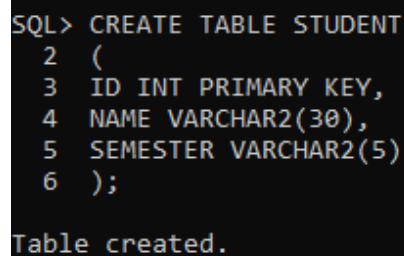
Figure 7: DROP TABLE statement

3 WORKING WITH PRIMARY KEYS

Primary keys are used to uniquely identify each row in a particular table. Primary key needs to be specified during the creation of the table. It can also be specified later by altering the table, but we will see the ALTER TABLE command in a later lab.

```
CREATE TABLE STUDENT
(
  ID INT,
  NAME VARCHAR2(30),
  SEMESTER VARCHAR2(5),
  PRIMARY KEY (ID)
);
```

```
CREATE TABLE STUDENT
(
  ID INT PRIMARY KEY,
  NAME VARCHAR2(30),
  SEMESTER VARCHAR2(5)
);
```



```
SQL> CREATE TABLE STUDENT
2  (
3  ID INT PRIMARY KEY,
4  NAME VARCHAR2(30),
5  SEMESTER VARCHAR2(5)
6  );
Table created.
```

Figure 8: Creating a table with a primary key

4 TIPS AND TRICKS

4.1 Checking current user

To check which user is currently logged in, write the following command:

```
SHOW USER;
```

4.2 Disconnect/Log out

While logged in, simply type the following:

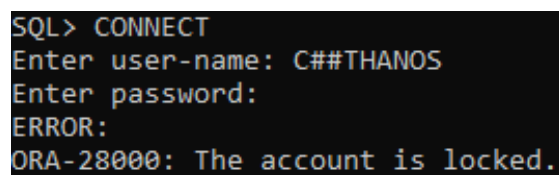
```
DISC;
```

Or,

```
DISCONNECT;
```

4.3 Locked account solution

You may come across an error that says that a particular account is locked.



```
SQL> CONNECT
Enter user-name: C##THANOS
Enter password:
ERROR:
ORA-28000: The account is locked.
```

Figure 9: Locked account error

In such case, do the following:

```
CONNECT SYS/SYS AS SYSDBA;
```

```
ALTER USER username IDENTIFIED BY correct_password ACCOUNT UNLOCK;
```

```
SQL> CONNECT SYS/SYS AS SYSDBA;
Connected.
SQL> ALTER USER C##THANOS IDENTIFIED BY INEVITABLE ACCOUNT UNLOCK;
User altered.
```

Figure 10: Unlocking an account

4.4 Saving your work

If you want to save your work that you did in a particular session, write the following before logging out:

```
COMMIT;
```

4.5 Clearing the console

In case you have a huge pile of text in your console and you want to clean things up, write the following:

```
CL SCR;
```
