**Purpose:**

In this week's lab, you will create users and **grant**  specific privileges to those users. You will also assign a default tablespace and create some tables in that tablespace owned by one of the new users. You will then create several new roles, grant privileges to the roles and then grant the roles to particular users based on the user's requirements for being able to update, select, or delete data in those tables.

**Deliverable:**

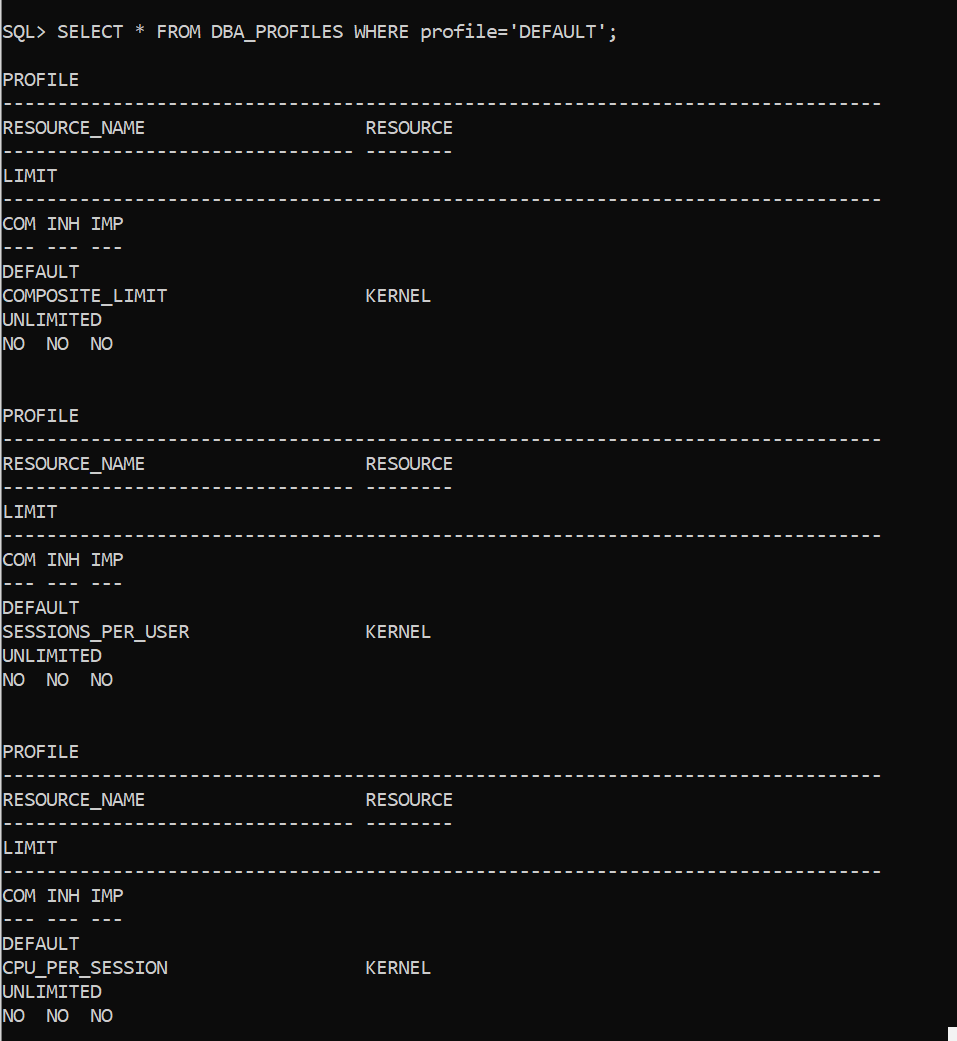
Complete this document, submit it via BrightSpace, and (if asked) demonstrate the results to the lab professor. This lab is worth 2 marks towards the lab/assignments component of your final grade.

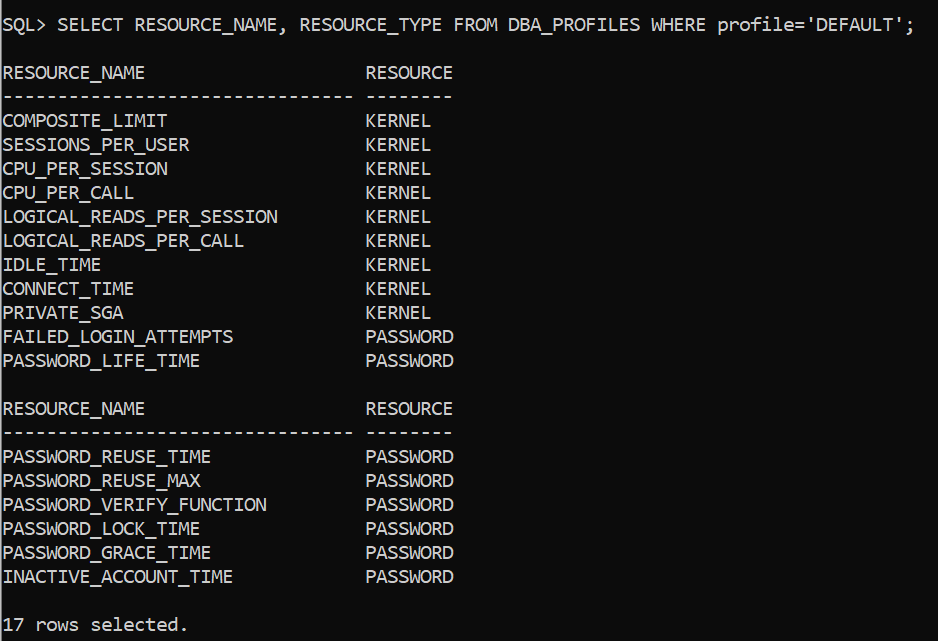
**Requirements:**

1. **Predefined Privileges:** Oracle classifies SYSTEM privileges as actions that apply to the system or types of objects. For example, CREATE DATABASE, CREATE TABLESPACE, and CREATE TABLE. The entire set of system and object privileges are listed and described in the documentation available at:  
   [Privileges (oracle.com)](https://docs.oracle.com/cd/E11882_01/timesten.112/e21642/privileges.htm#TTSQL338)   
   You will need to scroll down to see the long list of system privileges and could then navigate to each of their descriptions.
2. The GRANT syntax is described at:  
   [GRANT (oracle.com)](https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/GRANT.html#GUID-20B4E2C0-A7F8-4BC8-A5E8-BE61BDC41AC3)

Oracle classifies OBJECT privileges as actions that apply to a particular object. For example, SELECT, INSERT, UPDATE, and DELETE actions on a specific table. Additionally, the owner of an object – say I created a table named *Test -* would automatically have SELECT, INSERT, UPDATE, etc. rights on *Test*. A user can also be granted rights on an object owned by another user – we will do this later in the current lab.

1. **Start SQLPLus and logon as SYS AS SYSDBA.**
   1. How many predefined system privileges are listed in the system\_privilege\_map view: \_\_\_\_257 (SELECT COUNT(\*) FROM system\_privilege\_map;\_\_\_
   2. How many predefined object privileges are listed in the dba\_tab\_privs view: \_\_51616 (SELECT COUNT(\*) FROM dba\_tab\_privs;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Hint: You won't want to *SELECT \* FROM dba\_tab\_privs* as a large number of rows will be returned. Instead use an aggregation function to determine the number of rows in the view).
2. **Use the DBA\_PROFILES table to list the details of the ‘DEFAULT’ profile.**Provide a screen shot below:





List 2 aspects of the default profile that could lead to performance issues: \_\_COMPOSITE\_LIMIT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_SESSIONS\_PER\_USER\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

List 1 aspect of the default profile that could lead to security issues: \_\_PASSWORD\_REUSE\_TIME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

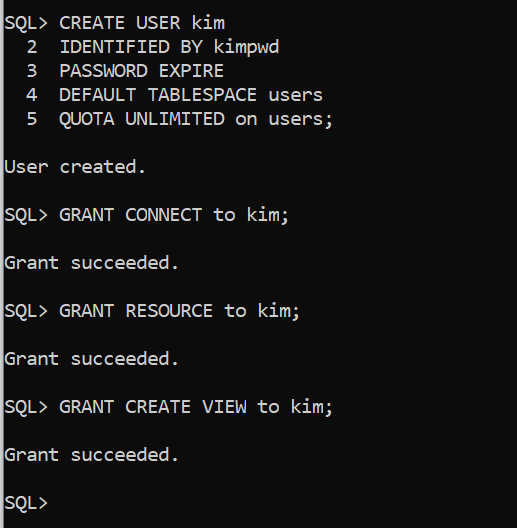
\* NOTE : DBA\_PROFILES

RESOURCE TYPE : KERNEL – PERFORMANCE

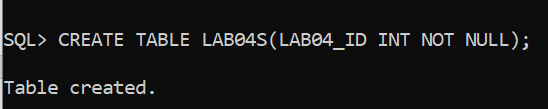
PASSWORD - SECURITY

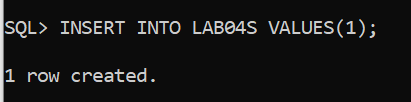
1. In a sequence of “create user ….”, “alter user …” and other privilege granting steps, you are to create a new user with the following criteria:
   1. Username - your last name (e.g., king)
   2. Password – your last name (e.g., kingpwd)
   3. **Set the password to be expired**. **(**It will need to be changed on next connection.)
   4. **Use the DEFAULT profile.**
   5. Use the default tablespaces (i.e., USERS and TEMP).
   6. Use the “GRANT …. TO ….” command multiple times to grant CONNECT, RESOURCE and CREATE VIEW to the new user.

Show your work below:



* 1. Keep your SQLPLUS session running as SYS AS DBA, and open a new command window (not as administrator). Then, logon to SQLPlus with your new account in the new window. You should get an 'expired password' message. When prompted enter *lastname*pwd as your “New” password. (It will let you reuse the original one….)
  2. Try to create a simple empty table from your new account. If successful, try to insert one record. What occurs? Show your work

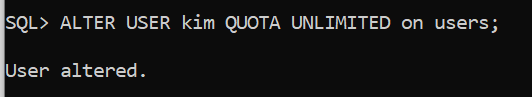




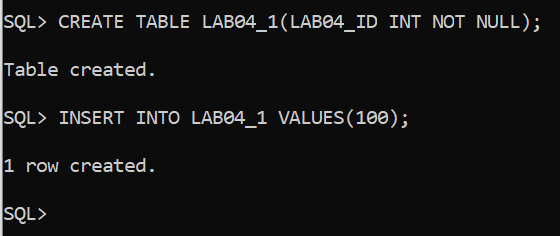
It created 1 row. Because when I create a new user, I already use QUOTA function.

It suppose to be can’t insert value, if I didn’t use QUOTA when I create a new user.

* 1. Keep your new account window open but change focus back to your original SYS as SYSBA window to alter the quota on the ‘USERS’ tablespace for your new user to be UNLIMITED.   
     e.g., “ALTER USER myname QUOTA UNLIMITED ON USERS;” Show your work here:



* 1. Return to your new account in the other window and retry creating a new table and inserting a single row. Show your work here:



1. **Creating Roles:** The information on the next page describes privileges that certain users have on specific tables. Technically, we could grant the correct object privilege on each table on a user-by-user basis … but that approach is inelegant, inefficient and scales poorly – what is feasible for the first 6 users is terribly problematic for the next 30,000. A more realistic (simplified) example would be BrightSpace – it has an ever-increasing number of users assigned to the *student* role and approximately 1,000 users assigned to the *instructor* role – with each role having different privileges.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object Privileges** | **Tables (Objects)** | | | |
| AUTHORS | AUTHOR\_TITLES | TITLES | PUBLISHERS |
| **S**ELECT | UserA, UserB,  UserC, UserD,  UserE | UserA, UserB,  UserC, UserD,  UserE | UserA, UserB, UserC, UserD,  UserE, UserF | UserA, UserB,  UserC, UserD,  UserE |
| **I**NSERT | UserA, UserB, | UserC, UserD,  UserE | UserF | UserC, UserD,  UserE |
| **U**PDATE | UserA, UserB, | UserC, UserD,  UserE | UserF | UserC, UserD,  UserE |
| **D**ELETE | UserA, UserB, | UserC, UserD,  UserE | UserF | UserC, UserD,  UserE |

* 1. Take the data from the table above and re-organize it in the format below. I have done the first row for you. Doing this will help you spot the privileges/object pairings for each user more easily. (Note: S=Select, I=Insert, U=Update, and D=Delete).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Authors** | | | | **Author\_Titles** | | | | **Titles** | | | | **Publishers** | | | |
|  | **S** | **I** | **U** | **D** | **S** | **I** | **U** | **D** | **S** | **I** | **U** | **D** | **S** | **I** | **U** | **D** |
| **UserA** | **Y** | **Y** | **Y** | **Y** | **Y** | N | N | N | **Y** | N | N | N | **Y** | N | N | N |
| **UserB** | **Y** | **Y** | **Y** | **Y** | **Y** | **N** | **N** | **N** | **Y** | **N** | **N** | **N** | **Y** | **N** | **N** | **N** |
| **UserC** | **Y** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** | **Y** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** |
| **UserD** | **Y** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** | **Y** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** |
| **UserE** | **Y** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** | **Y** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** |
| **UserF** | **N** | **N** | **N** | **N** | **N** | **N** | **N** | **N** | **Y** | **Y** | **Y** | **Y** | **N** | **N** | **N** | **N** |

* 1. What is the **minimum** number of roles (i.e., grouping by common requirements) required to assign the correct privileges to the appropriate users? 🡺 \_\_\_3\_\_\_\_\_\_.
  2. List the roles (e.g., role1, role2, etc.) and the users that should be assigned to each role.

Role1 : UserA, UserB

Role2 : UserC, UserD, UserE

Role3 : UserF

1. Use the template below to create 6 users named UserA, UserB, UserC, UserD, UserE and UserF, and to grant them connect privileges.

*CREATE USER* ***UserA*** *IDENTIFIED BY* ***usera****pswd*

*DEFAULT TABLESPACE users*

*TEMPORARY TABLESPACE temp*

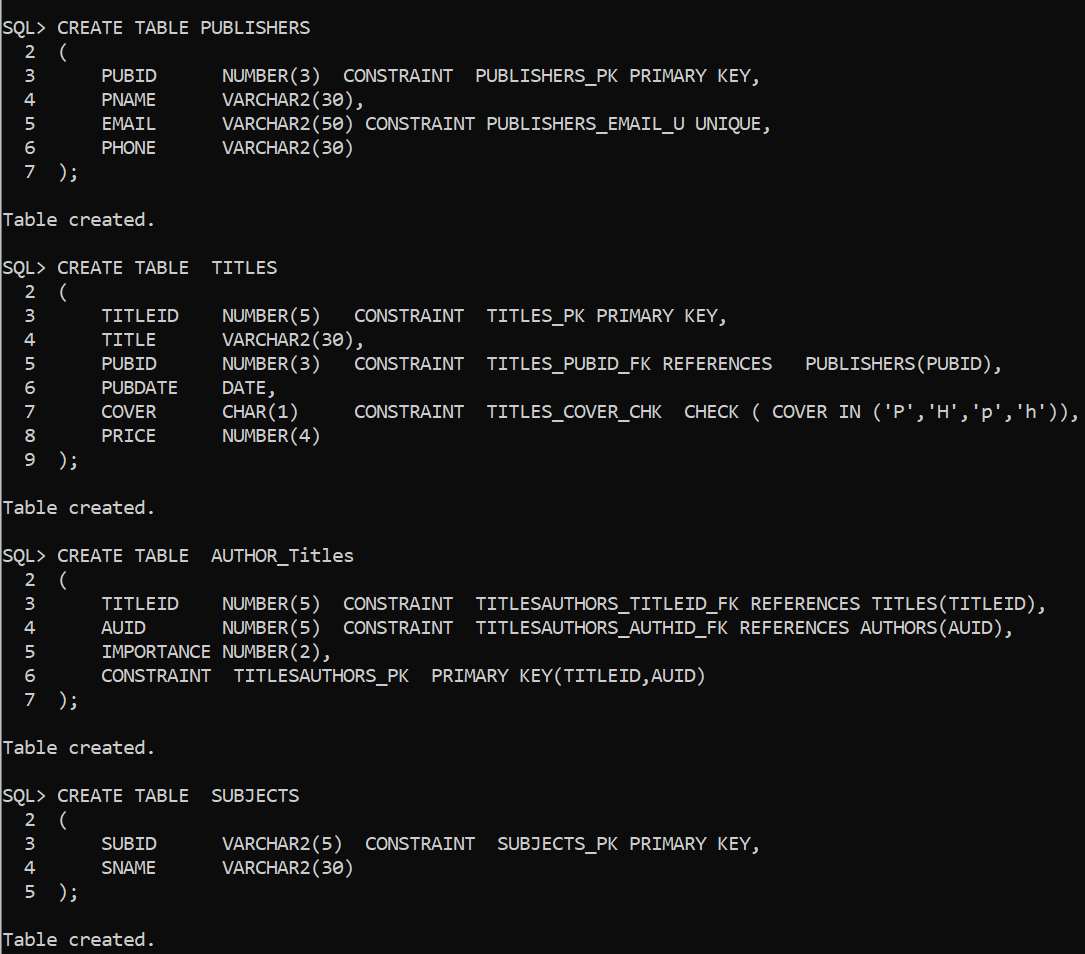
*QUOTA UNLIMITED ON users;*

*Grant connect to* ***usera****;*

**Can these users create a table? \_NO\_**

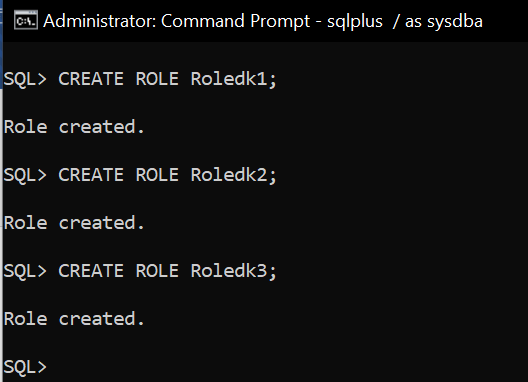
**Why? \_Because these users don’t have privilege to create new table\_\_\_\_**

1. Logon to your general user account (e.g., *king*). Use the SQL scripts posted in Brightspace create the *Authors, Author\_Titles, Titles,* and *Publishers* tables. Then, use the posted insert file to add data to the tables. Document your work below.





1. Logon as SYS AS SYSDBA and enter: *CREATE ROLE Rolefname1;* (for example, *Roleking1*).Create the number of roles you determined were needed in Question 6 above. Additional information on creating roles can be found here: <http://docs.oracle.com/database/121/SQLRF/statements_6014.htm#SQLRF01311>



1. Review the basic syntax for GRANTING an object privilege (e.g., SELECT, etc.) on a schema object (e.g., *king.authors*) to a role (e.g., Role*king*1). Read the **Restriction on Object Privileges** sentence under the **Grant\_Object\_Privileges** clause description as it contains important information regarding whether, for example, you can grant a select on king.authors and select on king.titles in the same grant statement. Granting these two privileges in two separate grant statements is fine. <http://docs.oracle.com/database/121/SQLRF/statements_9013.htm#SQLRF01603>

Copy your GRANTs here:

GRANT SELECT, INSERT, UPDATE, DELETE ON kim.authors TO Roledk1;

GRANT SELECT ON kim.publishers TO Roledk1;

GRANT SELECT ON kim.subjects TO Roledk1;

GRANT SELECT ON kim.titles TO Roledk1;

GRANT SELECT ON kim.author\_titles TO Roledk1;

GRANT SELECT ON kim.authors TO Roledk2;

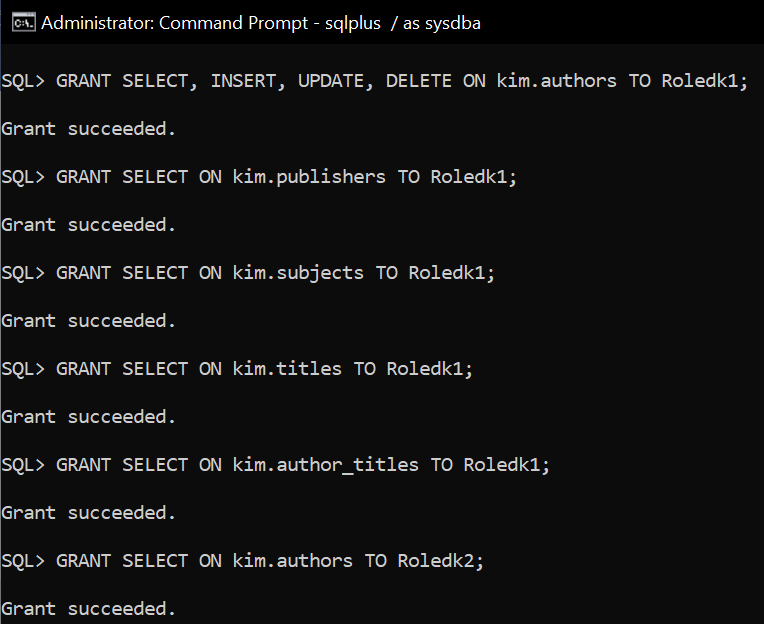
GRANT SELECT, INSERT, UPDATE, DELETE ON kim.publishers TO Roledk2;

GRANT SELECT ON kim.subjects TO Roledk2;

GRANT SELECT ON kim.titles TO Roledk2;

GRANT SELECT ON kim.author\_titles TO Roledk2;

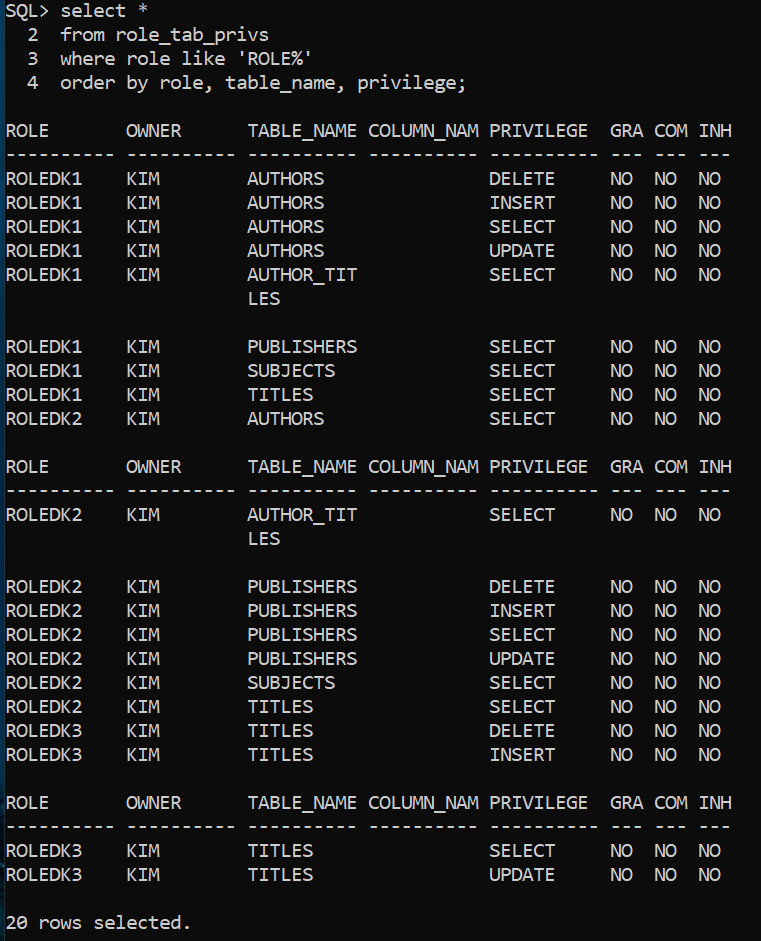
GRANT SELECT, INSERT, UPDATE, DELETE ON kim.titles TO Roledk3;



* 1. Enter: ***DESC role\_tab\_privs*** for a description of the structure of the view that contains information about roles and their privileges. You may notice the first 5 columns are unnecessarily wide for our data. Use the COLUMN / FORMAT command to modify the displayed column width. Use this example as a guide:

SQL> **column role format a10**

**Write a SQL query** that selects all columns from the ***role\_tab\_privs*** view for all of your newly created roles and orders the results by role, table\_name, and privilege. Paste the command and results here:



1. Our next step is to grant the appropriate role or roles to the correct user(s).

Granting a role to a user is very similar to granting a privilege to a role or, for that matter, granting a role to another role. Use 'g' above as a model for granting a role to a user. Paste the commands and results here:

GRANT Roledk1 to UserA;

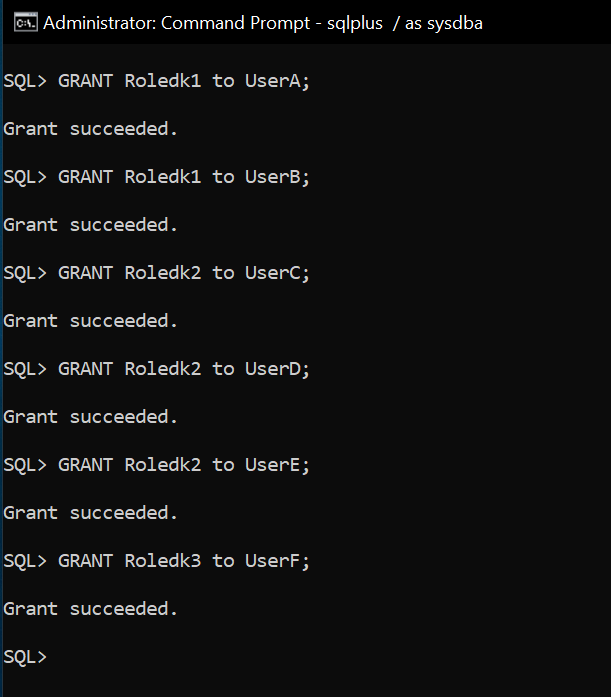
GRANT Roledk1 to UserB;

GRANT Roledk2 to UserC;

GRANT Roledk2 to UserD;

GRANT Roledk2 to UserE;

GRANT Roledk3 to UserF;



Additional detailed information is available in the GRANT section of the SQL Reference Guide <http://docs.oracle.com/database/121/SQLRF/statements_9013.htm#SQLRF01603>

* 1. After all of the role granting to users has completed, enter the following command and then paste the query and results in the space provided below:

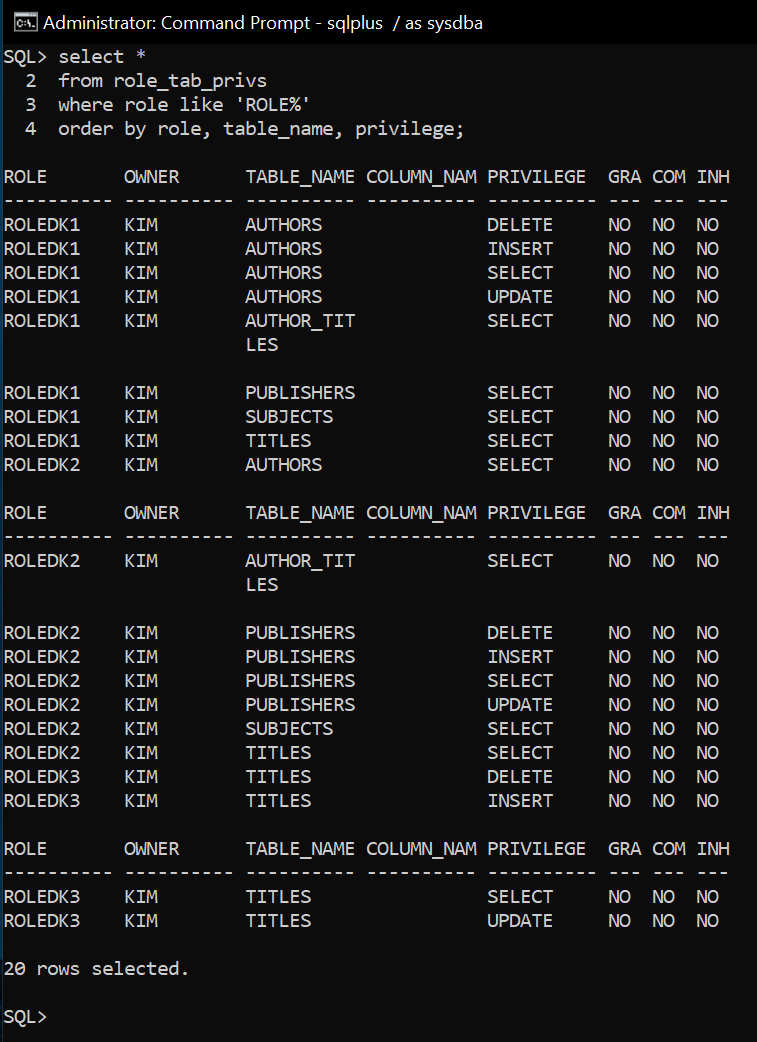
***select \****

***from role\_tab\_privs***

***where role like 'ROLE%'***

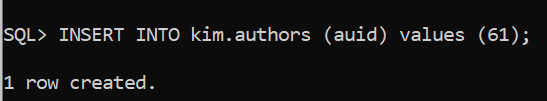
***order by role, table\_name, privilege;***

Paste the query and results here:

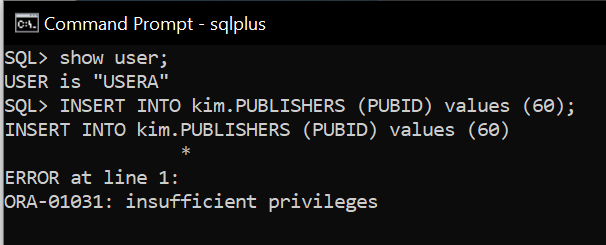


1. To demonstrate that the **GRANT**s were correctly established, write the query and show the results for the following actions. (Make sure your results are as expected based on the requirements originally given in the table in Step 6.)
   * + - 1. UserA inserting Author ID = 60 in Authors

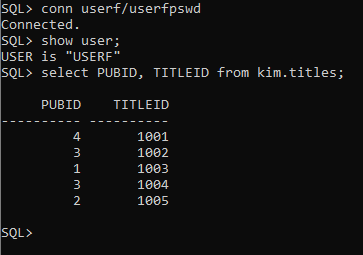




* + - * 1. UserA inserting Publisher ID = 60 in Publishers



* + - * 1. UserF selecting publisher ID and Title ID from Titles



**You’re done!**