

For each question, show the SQL statement(s) that you have used in order to address the question.

1. Create a test table called **JonesB_TestTable** (where “JonesB” represents your user name) and populate it with some sample data. [10]

```
CREATE TABLE JonesB_TestTable
(Field1# CHAR (5) NOT NULL,
Field2 CHAR (30) NOT NULL,
CONSTRAINT TestPK PRIMARY KEY (Field1));
```

// Note: Grader will test for the existence of the table by issuing the statements:

```
DESCR <UserSchema>.JonesB_TestTable;
SELECT * FROM <UserSchema>.JonesB_TestTable;
```

- 2a. Create a role and for its name, use the concatenation of your user account and the word “Role”. For instance, the role may be called **JobesB_Role** (where “JonesB” represents your user name). [02]

```
CREATE ROLE JonesB_Role;
```

Note: If the role account is created in the common database (CDB), it must be preceded with the characters C##, thus

```
CREATE ROLE C##_JonesB_Role;
```

- 2b. Grant to your role the system privileges that facilitate application development (see section 13.2.2 of your text). [11]

```
GRANT CREATE SESSION, CREATE TABLE, CREATE SEQUENCE, CREATE VIEW, CREATE ROLE, CREATE PROCEDURE,
CREATE TRIGGER, CREATE TABLESPACE, CREATE SYNONYM, INSERT ANY TABLE, ANALYZE ANY, SELECT ANY
DICTIONARY TO JonesB_Role;
```

Note: In the KSC environment, this will not work for CS360 students, because they do not have the DBA privilege to execute this command. However, the role Developer has been created and granted the above-mentioned privileges by the course instructor; moreover, students have been given privileges to this Developer role. So for the purpose of implementation of the CS360DB at KSC, the following statement may be issued by each student:

```
GRANT Developer TO JonesB_Role;
```

- 2c. Grant to your role, SUDI privileges on **JonesB_TestTable** (see section 13.2.3 of text for SUDI privileges). [04]

```
GRANT SELECT, UPDATE, DROP, INSERT ON JonesB_TestTable TO JonesB_Role;
```

- 2d. There is a catalog table that you can access to see all the system privileges granted to all roles in the system. Its name is **Role_Sys_Privs**. Study its structure, and then issue an appropriate SQL statement to display the system privileges associated with your role. Show the SQL statement used. [06]

```
SELECT Role, Privilege FROM Role_Sys_Privs WHERE Role LIKE 'JonesB_Role%' OR Role LIKE 'JONESB_ROLE%';
```

You might also want to take a look at the view called **Role_Role_Privs** and run a similar query.

3.

- 3a. Create a test user account called **JonesB_User** (where “JonesB” represents your user name). Assign the user to your tablespace; also use a password that you will remember. Your current tablespace name is **TBS_JonesB** (where “JonesB” represents your user name). [04]

```
CREATE USER C##_JonesB_User IDENTIFIED BY "TestUser_360"  
DEFAULT TABLESPACE TBS_ JonesB  
QUOTA UNLIMITED ON TBS_ JonesB;
```

Note: If the user account is created in the common database (CDB), it must be preceded with the characters **C##**.

- 3b. Try logging on to the database as **JonesB_User**. Record the result of your attempt, and provide an explanation for this observation. [04]

You should get the error ORA-01045, which states that the user JonesB lacks CREATE SESSION privilege. This is so because your test user account has not been granted appropriate privileges to access the database.

- 3c. Log on with your normal user account and grant your role (created in 2a) to user **JonesB_User**. [02]

```
GRANT JonesB_Role TO C##_JonesB_User;
```

- 3d. Try logging on as **JonesB_User** once more and record the result of your attempt. Provide an explanation for this observation. [04]

Log-on should now be allowed. This is so because in the previous exercise, appropriate privileges contained in your test role, were granted to your test user.

- 3e. Log on as **JonesB_User** and try running a query on your test table. To do this, you must either create synonym for **JonesB_TestTable** in the schema of **JonesB_User**, then run the query on the synonym, or you must qualify the table name when you run the query on **JonesB_TestTable**. Record the result of your attempt, and explain why you obtained that result. [10]

```
SELECT * FROM C##JonesB_User.JonesB_TestTable;
```

Alternately, run the following two statements:

```
CREATE SYNONYM JonesB_TestTable FOR C##_JonesB_User.JonesB_TestTable;  
SELECT * FROM JonesB_TestTable;
```

Access should be allowed. This is so because in 2c above, SUDI privileges were granted to your test role, which in then grant to your test user.

- 3f. Switch to your normal user account and revoke the SUDI privileges on **JonesB_TestTable** from your role. [04]

```
REVOKE SELECT, UPDATE, DROP, INSERT ON JonesB_TestTable FROM C##_JonesB_User_Role;
```

- 3g. Log on as **JonesB** again and try running a query on your **JonesB_TestTable**. Record the result of your attempt. Provide an explanation for this result. [06]

Access should now be denied, since the SUDI privileges on **JonesB_TestTable** have been removed. The exact message is:

ORA-01031: insufficient privileges

4.

Study your music database that you created for the previous assignment, and reflect on what you have learned about the system catalog. Write SQL statements to show what the contents of the system catalog tables USER_TABLES (attributes Table_Name and Tablespace_Name) and USER_TAB_COLUMNS (attributes Column_Name and Table_Name) would be. Include a screen shot of the query results. [10]

Contents of USER_TABLES:

```
SELECT table_name, tablespace_name FROM user_tables;
```

```
// This statement would produce the following result:
```

<u>Table Name</u>	<u>Tablespace Name</u>
COUNTRIES	TBS_<UserName>
MUSICIANS	TBS_<UserName>
COMPOSITIONS	TBS_<UserName>
ENSEMBLES	TBS_<UserName>
PERFORMANCES	TBS_<UserName>
ENSEMBLE_MEMBERS	TBS_<UserName>

```
// Note: <UserName> represents the name of the user who owns the table (each student has an account).
```

Contents of USER_TAB_COLUMNS:

```
SELECT column_name, table_name FROM user_tab_columns;
```

```
// This statement will produce a list with the following appearance:
```

<u>Column Name</u>	<u>Table Name</u>
CNTRYCD	COUNTRIES
CNTRYNAME	COUNTRIES
MNO	MUSICIANS
MNAME	MUSICIANS
DOB	MUSICIANS
MCOUNTRY	MUSICIANS
CNO	COMPOTITIANS
TITLE	COMPOTITIANS
MNO	COMPOTITIANS
CDATE	COMPOTITIANS
ENO	ENSEMBLES
ENAME	ENSEMBLES
ECOUNTRY	ENSEMBLES
MNO-MGR	ENSEMBLES
ENO	ENSEMBLE_MEMBERS
MNO	ENSEMBLE_MEMBERS
INSTRUMENT	ENSEMBLE_MEMBERS
PNO	PERFORMANCES
PDATE	PERFORMANCES
CNO	PERFORMANCES
CITY	PERFORMANCES
PCOUNTRY	PERFORMANCES
ENO	PERFORMANCES