DBA 625 -- Assignment #2, Due by Wednesday, April 18th by 7pm

Printouts only, No E-mails please !

Before starting to work on this assignment you must be sure that your instance is **running** and that you have these Tablespaces (SYSTEM, SYSAUX, USERS, UNDOTBS1, BLUE, RED and INDX) located in directories like required for your Assignment #1. If Tablespace INDX is NOT created, then make its file 2M large, without auto-extension and with system allocated extents. Number of other Permanent and Temporary Tablespaces is NOT relevant for this task. You must NOT have DAVE's or LARA'S accounts created yet. Number of other users and profiles is also not relevant here.

Finally connect to your instance as SYSDBA. Then record your Unix sessions as ***"Assig2.doc"*** spool document. You have to do required steps in **a sequence**, because the next step depends on the previous one! The spool file must reflect **COMMAND LINE** syntax. After finishing all steps, you are going to provide just this **spool document**.

One:

a) Create new user called DAVE and assign RED and TEMP tablespaces to him, also DEFAULT profile. He should be allowed to use only 2M of RED tablespace, and also 1M of INDX tablespace.

b) Create another user called LARA and assign BLUE and TEMP tablespaces to her, also DEFAULT profile, but no quota initially for her and her account should be locked. Both users should get role that will allow them to connect to SQL and individual privilege for creating tables.

c) Create new profile called ITPROF so that:

- One session per user is allowed
- Idle CPU time is no more than 1 hour
- Three false logins are allowed and that account should remain locked for 2 minutes after the fourth false login

d) Assign this profile now to DAVE and enable all restrictions for profile ITPROF. Then connect to SQL*PLUS as DAVE and try to connect to SQL*PLUS as DAVE again (in a second session). What happened?

e) As SYSTEM modify profile ITPROF so that only two false logins are allowed and that password life time is only 2 months with the grace period of 5 days. Then try to login as DAVE with the wrong password 3 times. Wait only 1 minute and provide now the right password. What happened and how can you rectify this problem → show both methods?

f) As SYSTEM try to create replica of SCOTT's table EMP in both DAVE's and LARA's accounts. Was it successful in both cases? How can you fix this problem? After doing that, verify that LARA (after login) may access her table EMP.

g) By joining 2 dictionary views display for users DAVE and LARA their account status, when the account will expire, what profile and tablespaces are assigned and what the current and maximal byte situation is for these tablespaces?
h) Now give individual privilege to DAVE, so that he can browse and add data in a table in any account and also that he can continue to give this privilege. Then connect as DAVE and give the same privilege to LARA?
i) As SYSTEM, check the appropriate dictionary view and observe only system privileges for those two users (show only relevant columns from this view)
j) Remove the privilege given to DAVE in h), then connect as LARA and try to ADD one row into SCOTT's table EMP.
Was it successful and why? Then repeat step i) and explain what is different now?
k) In DB Express display situation for DAVE's account firstly and then for LARA's account as two different pages.

Two:

Firstly run the script ***cr_orders.sql*** as user DAVE (you will need to adjust this script regarding Tablespace name here)
a) As DAVE, create a PK constraint for table CUSTOMERS, so that its checking can be delayed (till saving) LATER in the future, but for now it will behave like the non-delayed one. The state should be set to check only incoming data. Also add PK constraint for table ORDERS, so that its checking can be delayed PROMPTLY after its creation, while its state should be set to check both existing and incoming data. Related indexes should go to Tablespace INDX.
b) Also add an UK constraint on column NAME in table CUSTOMERS, so that will have default value for either its state or mode, and sub-default for the other one. Think hard here. You may want to check question g) below in order to make a right choice.
The related index will also go to tablespace INDX.
c) Create a FK constraint for table ORDERS so that might be LATER manually delayed (at save time). This constraint state follows the default value.
d) Then declare a CK constraint with condition that date of delivery may not be later than two weeks from the order date and also not before it. This constraint's mode is the default one, while the state follows the sub-default value.
e) As SYSTEM, join two most important constraint dictionary views to display the following: constraint name, type, status, validation, can be deferred or not, is it currently deferred, table name and what column(s) are involved for tables CUSTOMERS and ORDERS in DAVE's account.
f) As DAVE, try to make a short DML script (**yours and original)**, that will show how it is possible to insert a child record before its parent record, if the child FK constraint checking is delayed till save time.
Manually perform the change that will allow this check delay.
g) Then make another short script that will show how it is possible to enter same names for different customers (three), if the appropriate constraint is turned off at the creation time.
h) Now try to set the UK constraint in DAVE's table CUSTOMERS to the default state. What happened?
i) Then enable it so that it does NOT check the existing data. If not possible, fix this obstacle by creating a new object in DAVE's schema.

j) Next, try to enter a row with an existing customer name. What happened?
k) Perform the five step recipe for consolidating the situation with the Name column in DAVE's CUSTOMERS table (get rid of all customers with duplicate names by modifying duplicates).

You should download first the script **utlexcpt.sql** in order to create the 'container' table (if not done before).

Three:

You should create folders BACKUP and ARCHIVE and turn ON archiving here (if you did not do it so far), like shown in Session15.

Also, delete in RMAN all previous backup sets (if you have them) by using syntax **DELETE BACKUPSET number**. Then you may start this part.

a) Configure following options in RMAN

- Number of copies before becoming obsolete is 2
- Turn on auto backup of Control File
- Default folder to hold backup sets is **/home/oracle/BACKUP** and default name for these sets is **hot_%u_%s_%p**
- Exclude all MGMT tablespaces from every backup

b) Perform WHOLE, FULL and COLD database backup. This one may be used later as a base for Incremental backups. Allow only 4 files per set (peace)

Now show all backup sets created so far

c) Perform HOT and Incremental-Cumulative backup of tablespaces MINE and RED

Then show all backups of tablespace MINE

d) Perform HOT and Incremental-Differential backup of datafile that belongs to tablespace USERS. Then show all backups of that datafile

e) Display all files that are due for backup regarding retention policy established in a)

f) Perform HOT and Incremental-Cumulative backup of database

g) Display all backup sets that are beyond retention policy established in a). Then delete all these sets.

h) Show the content of **/home/oracle/BACKUP** folder

And try to have some fun !!!