COMP-1802-M01-2024-25

Advanced Database Technologies

Cornateanu Laurentiu
ID: 001295276
@mail: lc9754t@gre.ac.uk
VM : lc9754t
Uid: "C:\Program Files (x86)\VMware\VMware Remote Console\vmrc.exe" vmrc://classroom-vc.gre.ac.uk/?moid=vm-31423

Module Leader: Dr. Margarita Razgon

e-mail: m.razgon@gre.ac.uk

Abstract

This task demonstrates implementing a merchant database's administration and control practices using Oracle SQL*Plus. The main goals include producing users, assigning appropriate permissions, designing /administrating tablespaces and applying constraints to ensure data integrity.

The goal was accomplished as part of a complete case study, simulating reality challenges faced by a database administrator (DBA). Detailed necessities addressed in this assignment include:

- Generate and organize database users with distinct tablespaces, profiles and password policies.
- Implement user access control based on departmental needs, such as read-only or read-write privileges to specific tables.
- Design and populate a new table with appropriate constraints to ensure data accuracy and reliability.
- Handling potential DBA issues such as recovering abandoned schemas and troubleshooting common SQL errors such as ORA-00942: table or view does not exist.

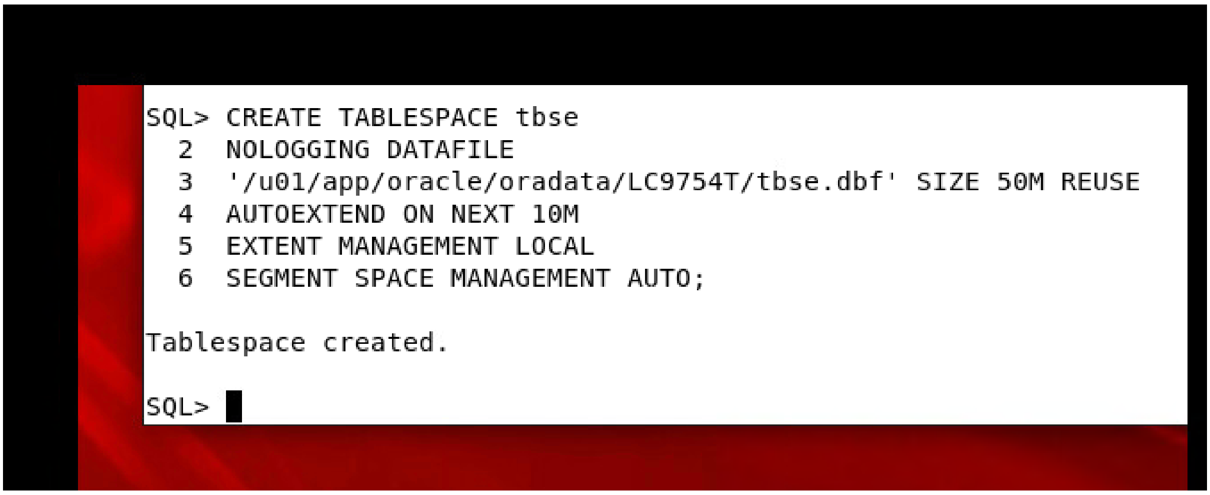
Supported by SQL commands and explanations of their implementation. In addition, alternatives and their potential benefits and drawbacks are explored where relevant.

Contents

1.Tablespace

1. Please create a new permanent tablespace tbse in your database to store additional information that can be updated on a regular basis. This tablespace should have an initial size of 50 megabytes and one data file. The space in this tablespace should be managed locally and extended automatically by 10 megabytes.

```
CREATE TABLESPACE tbse
DATAFILE '/path to database/tbse.dbf ' SIZE 50M
AUTOEXTEND ON NEXT 10M
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```



```
SQL> CREATE TABLESPACE tbse
  2  NOLOGGING DATAFILE
  3  '/u01/app/oracle/oradata/LC9754T/tbse.dbf' SIZE 50M REUSE
  4  AUTOEXTEND ON NEXT 10M
  5  EXTENT MANAGEMENT LOCAL
  6  SEGMENT SPACE MANAGEMENT AUTO;

Tablespace created.

SQL> 
```

Fig 1.

- *Initializing 50 MB, in my point of view, is a moderate extra data with no more resources provision.*
- *Using AUTOEXTEND I am shore about the storage expansion to sys define limits and in this way we don't have to do it manually in case of need it.*

- *To improve performance: Extent Management with Local and efficiency of space.*
- *What I construct here has problems, including the possibility of runaway storage if it is not observed; because of that, I added an extra layer at position 6.*
- *This set optimizes space. For a better translation*, talk with Oracle to manage space routinely.

2. User Creation

You are required to create the following two new users with login requirements specified below:

a) One user called cust for the Customer Service department, and one user called invent for the Inventory department.

b) The default tablespace with an unlimited quota for both users should be set to tbse (created in Question 1). The temporary tablespace for both users should be set to temp. The profile profdep should be assigned to both users.

c) The user cust should be forced to change their password on the first login. The user invent does not have this requirement.

d) Both users should be forced to change their passwords every 2 months.

First of all, to implement all reuierments, we have to implement a few more steps before user creation:
Produce department of **CUSTOMER SERVICE**
Produce department of **INVENTORY**
Produce profile **PROFDEP**
***(FOR ALL THIS IS NOT REQUIERMENTS TO PRODUCE CODE AND TALK ABOUT IT)**

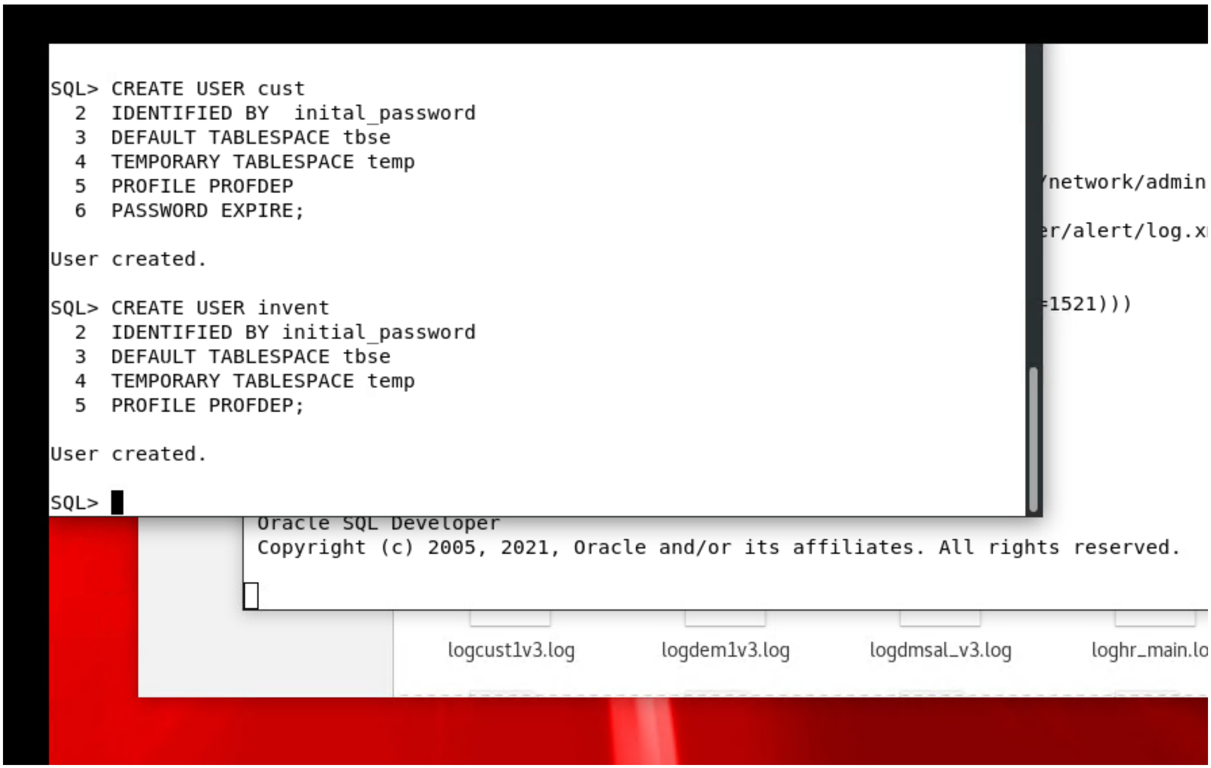
```
CREATE USER cust
IDENTIFIED BY initial_password
DEFAULT TABLESPACE tbse
TEMPORARY TABLESPACE temp
PROFILE profdep
PASSWORD EXPIRE;
```

- DEFAULT TABLESPACE tbse: Assigns the tbse tablespace with unlimited quota as the default tablespace for the user.
- TEMPORARY TABLESPACE temp: Assign the temp tablespace as the temporary
- Assigns the profdep profile.
- PASSWORD EXPIRE: Forces the user to change their password on the first login.
- Initialization expires password on first-time login improves security.

- Bring into line with management space and a better organization of data storing data in tbse by default tablespace.

CREATE USER invent
IDENTIFIED BY initial_password
DEFAULT TABLESPACE tbse
TEMPORARY TABLESPACE temp
PROFILE profdep;

- The only difference is that the user does not include PASSWORD EXPIRE, not forced to change the password the first-time login.



```
SQL> CREATE USER cust
  2  IDENTIFIED BY  inital_password
  3  DEFAULT TABLESPACE tbse
  4  TEMPORARY TABLESPACE temp
  5  PROFILE PROFDEP
  6  PASSWORD EXPIRE;

User created.

SQL> CREATE USER invent
  2  IDENTIFIED BY initial_password
  3  DEFAULT TABLESPACE tbse
  4  TEMPORARY TABLESPACE temp
  5  PROFILE PROFDEP;

User created.

SQL>
```

```
/network/admin
er/alert/log.x
=1521)))
Oracle SQL Developer
Copyright (c) 2005, 2021, Oracle and/or its affiliates. All rights reserved.
```

Fig.2.

3 Granting rights.

Two new users (created in Question 2) should have the following access rights:
a) Both users should be able to log in to the database.
b) The user cust should have Read/Only access to the OE.ORDERS
and OE.CUSTOMERS tables.
c) The user invent should have Read/Write access to the OE.WAREHOUSES table.
d) Both users should be allowed to create new tables and views.

GRANT CREATE SESSION TO cust;
GRANT CREATE SESSION TO invent;

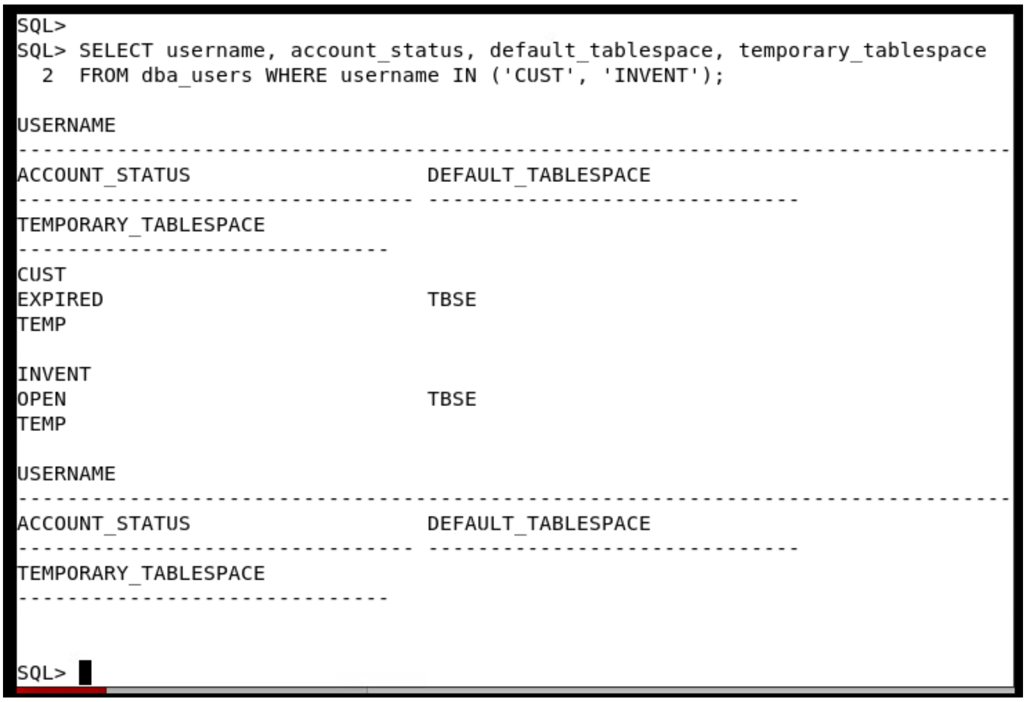
GRANT SELECT ON oe.orders TO cust;
GRANT SELECT ON oe.customers TO cust;

GRANT SELECT, INSERT, UPDATE, DELETE ON oe.warehouses TO invent;

GRANT CREATE TABLE, CREATE VIEW TO cust;
GRANT CREATE TABLE, CREATE VIEW TO invent;

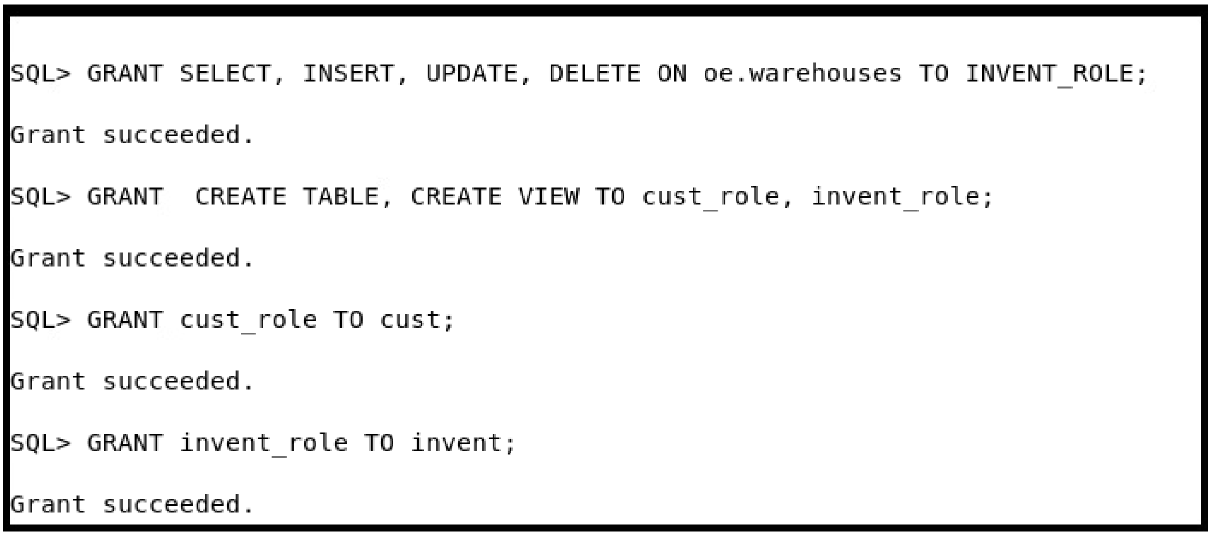
- I am not going beyond the role and tasks; I have precise authorizations and am in agreement with the role.
- Access read-only for cust user for accidental change.
- Invent users have full access to the OE.WAREHOUSES. More flexibility in inventory management.

Aids: improves safety.
Drawback: Endless continuing management of privileges.



```
SQL>
SQL> SELECT username, account_status, default_tablespace, temporary_tablespace
  2  FROM dba_users WHERE username IN ('CUST', 'INVENT');

USERNAME
--------------------------------------------------------------------------------
ACCOUNT_STATUS                   DEFAULT_TABLESPACE
-------------------------------- ------------------------------
TEMPORARY_TABLESPACE
------------------------------
CUST
EXPIRED                          TBSE
TEMP

INVENT
OPEN                             TBSE
TEMP

USERNAME
--------------------------------------------------------------------------------
ACCOUNT_STATUS                   DEFAULT_TABLESPACE
-------------------------------- ------------------------------
TEMPORARY_TABLESPACE
------------------------------


SQL> 
```

Fig.3.



```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON oe.warehouses TO INVENT_ROLE;

Grant succeeded.

SQL> GRANT  CREATE TABLE, CREATE VIEW TO cust_role, invent_role;

Grant succeeded.

SQL> GRANT cust_role TO cust;

Grant succeeded.

SQL> GRANT invent_role TO invent;

Grant succeeded.
```

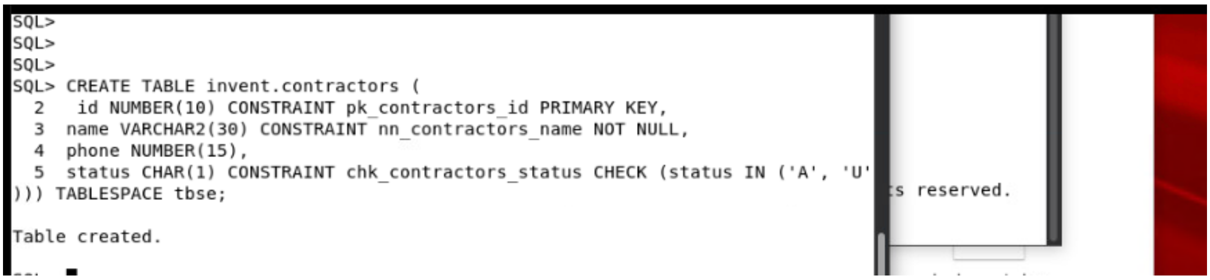
Fig.4.

4. Create a table

Create a new table Contractors in the schema of the user invent (created in Question 2) and stored in the tbse tablespace (created in Question 1).
Please use the structure described in the table below.
Make sure all the constraints have user-defined names.

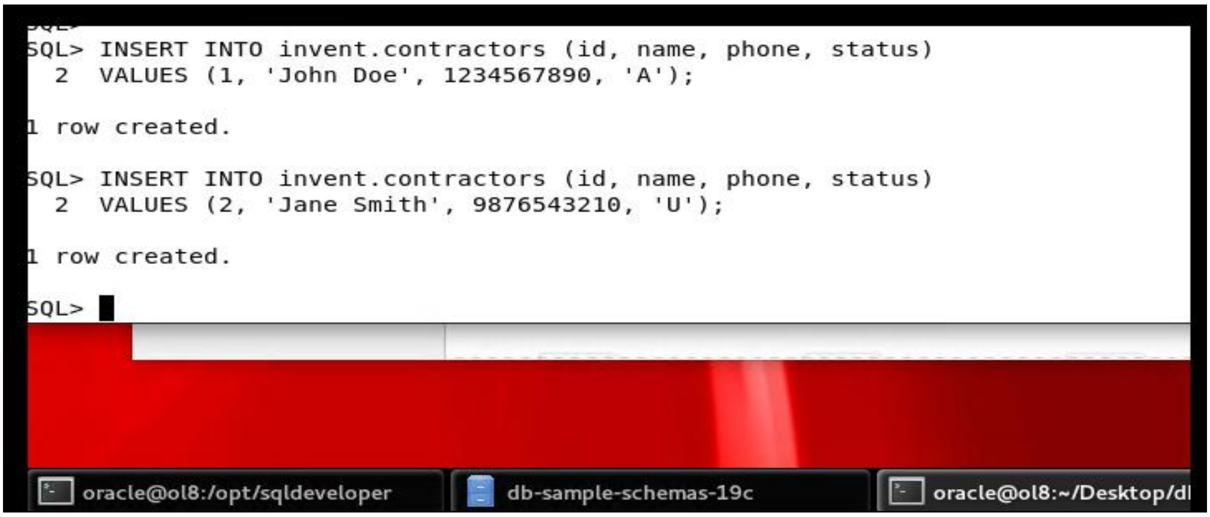
Column name	Data type	Constraint
id	NUMBER(10)	Primary Key
name	VARCHAR2(30)	Not Null
phone	NUMBER(15)	
status	CHAR(1)	Check constraint: this column should contain only two values: 'A' (for 'Available') and 'U' (for 'Unavailable')

CREATE TABLE invent.contractors (
d NUMBER(10) CONSTRAINT contractors_pk PRIMARY KEY,
name VARCHAR2(30) CONSTRAINT contractors_name_nn NOT NULL,
phone NUMBER(15),
status CHAR(1) CONSTRAINT contractors_status_ck CHECK (status IN ('A', 'U'))) TABLESPACE tbse;



```
SQL>
SQL>
SQL>
SQL> CREATE TABLE invent.contractors (
  2   id NUMBER(10) CONSTRAINT pk_contractors_id PRIMARY KEY,
  3  name VARCHAR2(30) CONSTRAINT nn_contractors_name NOT NULL,
  4  phone NUMBER(15),
  5  status CHAR(1) CONSTRAINT chk_contractors_status CHECK (status IN ('A', 'U'
))) TABLESPACE tbse;

Table created.
```

Fig.5.



```
SQL> INSERT INTO invent.contractors (id, name, phone, status)
  2  VALUES (1, 'John Doe', 1234567890, 'A');

1 row created.

SQL> INSERT INTO invent.contractors (id, name, phone, status)
  2  VALUES (2, 'Jane Smith', 9876543210, 'U');

1 row created.

SQL> █
```

Fig.6.

5. Recovery.

You have accidently dropped an important schema (for example, hr) in your database. Please discuss the best functionality to use in order to recover it quickly and safely, providing SQL commands and explanations of the steps needed to recover data. One more thing what is important , recovery mode is different depince what kind of database system is use.

Use Flashback

- Suitable for a comprehensive recovery but requires backup management.
- Benefits: Flexibility with Flashback, full recovery with RMAN.
- Drawbacks: Flashback must be pre-configured; RMAN requires additional storage.
- You can try to shut down Oracle sometimes really works.

Check in the recycle bin

```
SELECT object, name, type
FROM recyclebin;
```

Table restore

```
FLASHBACK TABLE hr.employees TO BEFORE DROP;
```

6. ORA-00942 Error

Possible Reasons and Solutions:
Reason: Table/view does not exist.
Solution: Verify the object name and schema.

- SELECT * FROM all_tables WHERE table_name = 'MYTABLE';
- Reason: Lack of privileges.
- Solution: Grant required privileges.
- GRANT SELECT ON schema.mytable TO user;

Reason: Incorrect schema context.
Solution: Use fully qualified names.

- SELECT * FROM schema.mytable;
- Benefits: Comprehensive troubleshooting ensures resolution.
- Drawbacks: Time-intensive for larger databases.

I just give some exemple with the code specification . I will discuss a little bit about the possibilities to can try : Look in the recycle bin what is appling for flashback, or directly to recycle bin.Ussulay, the SQL have a back an,d if you don't want to waste your time and investigate farword, just put it back from the back up.Now we can disscusing about transaction log where you can use point in back time to recovery LOG Based. We have on hand a tool what use the third party ApexSQL Restore.IN all this cases we don't have to take all precaution. Act quickly, validate restore data, establish redundancy.

Journals and Research Papers:

- ACM Digital Library (Search for "Database Recovery Techniques")
- IEEE Xplore (Search for "Schema Recovery in Databases")

Cornateanu Laurentiu