

Table of Contents

--Lesson 1. Data Dictionary Workshop	2
--Lesson 2. Architecture / Files Workshop	6
--Lesson 3. Architecture / Processes Workshop	8
--Lesson 4. Architecture / Memory areas Workshop	9
--Lesson 5. Administrative Users Workshop	10
--Lesson 6. Startup Basics Workshop	12
--Lesson 6. STARTUP MOUNT Workshop	13
--Lesson 6. Shutdown Workshop	16
--Lesson 7. Server Parameters Files Workshop	17
--Lesson 8. User Basics Workshop	19
--Lesson 8. User Quota & Account Status Workshop	21
--Lesson 9. User Passwords Workshop	23
--Lesson 10. Auditing Workshop	25
--Lesson 11. System Privileges Workshop	27
--Lesson 11. Object Privileges Workshop	29
--Lesson 11. Roles Workshop	30
--Lesson 12. Tablespaces Basics Workshop	33
--Lesson 12. Tablespaces Workshop	34
--Lesson 13. Table Create Workshop	37
--Lesson 13. CTAS Workshop	38
--Lesson 13. DBMS_STATS Workshop	39
--Lesson 13. Table Compression Workshop	40
--Lesson 14. Indexes Workshop	42
--Lesson 14. Composite Index Workshop	44
--Lesson 15. Constraints Workshop	45
--Lesson 15. Constraint Exceptions Workshop	47
--Lesson 15. Constraint Indexes Workshop	48
--Lesson 16. Automatic Undo Management Workshop	49
--Lesson 17. Archiving Workshop	51
--Lesson 17. Multiple Archive Logs Workshop	52
--Lesson 17. Add Redo Log Workshop	54
--Lesson 18. Add Control File Workshop	56
--Lesson 18. Optional Create Control File Workshop	57
--Lesson 19. RMAN Workshop	59
--Lesson 20. Logical Backups Workshops	63
--Lesson 21. SQL*Loader Workshop	66
--Lesson 22. Oracle Networking Workshop	68

--Lesson 1. Data Dictionary Workshop

```
-- SkillBuilders Oracle9i Database Administration

-- Workshop Solution: A DBA's Introduction to Data Dictionary

-- 1. Connect to SQL*Plus with the SYSTEM username.

connect system/dave
-- Connected.

-- 2. Display the username and account status of all users on your database.

select username, account_status
from dba_users;

-- USERNAME                                ACCOUNT_STATUS
-- -----
-- SYS                                       OPEN
-- SYSTEM                                    OPEN
-- DBSNMP                                    OPEN
-- RMAN                                       OPEN
-- ANONYMOUS                                 OPEN
-- HR                                         OPEN
-- SCOTT                                     OPEN
-- U1                                        OPEN
-- DAVELAPTOP\Administrator                 OPEN
-- TEST                                      OPEN
-- Administrator                            OPEN
-- DAVE                                       OPEN
-- DEMO                                       OPEN
-- SH                                         OPEN
-- OUTLN                                    EXPIRED & LOCKED
-- WMSYS                                    EXPIRED & LOCKED
-- WKSYS                                    EXPIRED & LOCKED
-- ODM                                       EXPIRED & LOCKED
-- OLAPSYS                                  EXPIRED & LOCKED
-- QS_CB                                    EXPIRED & LOCKED
-- QS_CBADM                                 EXPIRED & LOCKED
-- QS_OS                                    EXPIRED & LOCKED
-- QS_ES                                    EXPIRED & LOCKED
-- QS_WS                                    EXPIRED & LOCKED
-- QS                                        EXPIRED & LOCKED
-- QS_ADM                                   EXPIRED & LOCKED
-- PM                                        EXPIRED & LOCKED
-- OE                                        EXPIRED & LOCKED
-- QS_CS                                    EXPIRED & LOCKED
```

```

-- ODM_MTR                EXPIRED & LOCKED
-- WKPROXY                EXPIRED & LOCKED
-- ORDSYS                 EXPIRED & LOCKED
-- ORDPLUGINS             EXPIRED & LOCKED
-- MDSYS                  EXPIRED & LOCKED
-- CTXSYS                 EXPIRED & LOCKED
-- XDB                    EXPIRED & LOCKED

-- 3. Display the name and status of all tablespaces on your database.

select tablespace_name, status
from dba_tablespaces;

-- TABLESPACE_NAME          STATUS
-- -----
-- SYSTEM                    ONLINE
-- UNDOTBS1                   ONLINE
-- TEMP                       ONLINE
-- CWMLITE                    ONLINE
-- DRSYS                      ONLINE
-- EXAMPLE                    ONLINE
-- INDX                       ONLINE
-- ODM                        ONLINE
-- TOOLS                      ONLINE
-- USERS                      ONLINE
-- XDB                        ONLINE

-- 11 rows selected.

-- 4. For each data file on your database, display the tablespace name, file
name, number of bytes used and number of blocks -- used.  If you're familiar
with the SQL*Plus BREAK and COMPUTE commands, display the sum of the BYTES
column - this is the
-- size of your database (excluding logs, control files and parameter files).

break on report
compute sum of bytes on report

select tablespace_name, file_name, bytes, blocks
from dba_data_files
/

-- TABLESPACE_NAME          FILE_NAME
BYTES      BLOCKS
-- -----
-- SYSTEM                    C:\ORACLE\ORADATA\DAVE\SYSTEM01.DBF
587202560    71680
-- UNDOTBS1                   C:\ORACLE\ORADATA\DAVE\UNDOTBS01.DBF
429916160    52480

```

```

-- CWMLITE                                C:\ORACLE\ORADATA\DAVE\CWMLITE01.DBF
10485760      1280
-- DRSYS                                  C:\ORACLE\ORADATA\DAVE\DRSYS01.DBF
10485760      1280
-- EXAMPLE                                C:\ORACLE\ORADATA\DAVE\EXAMPLE01.DBF
173015040    21120
-- INDX                                   C:\ORACLE\ORADATA\DAVE\INDX01.DBF
10485760      1280
-- ODM                                    C:\ORACLE\ORADATA\DAVE\ODM01.DBF
10485760      1280
-- TOOLS                                  C:\ORACLE\ORADATA\DAVE\TOOLS01.DBF
10485760      1280
-- XDB                                    C:\ORACLE\ORADATA\DAVE\XDB01.DBF
39976960     4880
-- USERS                                  C:\ORACLE\ORADATA\DAVE\O1_MF_USERS_ZMNKR400_.DBF
524288000    64000
--
-----
-- sum
1806827520

-- 10 rows selected.

-- 5. Use a V$ view to determine the name of the database you are connected to.

select name from v$database;

-- NAME
-- -----
-- DAVE

-- 6. Use a V$ view to determine the name of the instance you are connected to.

select instance_name from v$instance;

-- INSTANCE_NAME
-- -----
-- dave

-- 7. What is the current value of the background_dump_dest parameter?

show parameter background_dump_dest

-- NAME                                TYPE                VALUE
-- -----
-- background_dump_dest                 string              C:\oracle\admin\dave\bdump

select name, value
from v$parameter
where name = 'background_dump_dest'

```

```
/

-- NAME
-----
-- VALUE
-----
---
-- background_dump_dest
-- C:\oracle\admin\dave\bdump

-- 8. Locate the Alert file and view its contents.
-- Use a standard text editor to open file
-- c:\oracle\admin\dave\bdump>alert_dave.log
-- (your file name will differ based on your
-- background_dump_dest and your instance name)

--
```

--Lesson 2. Architecture / Files Workshop

```
-- SkillBuilders Oracle9i Database Administration class workshop solutions

--
-- Workshop:  Architecture / Files

-- 1. TRUE or FALSE:  The database cannot be accessed until an instance is
started.
-- Answer: TRUE

-- 2. Name the primary types of operating system files that make up an Oracle
database.
-- Answer: data files, redo log files, temp files, control files, parameter
files and password files

-- Use SQL*Plus to answer the following questions:

-- 3. What is the database name?
select name from v$database;

-- 4. What is the instance name?
select instance_name from v$instance;

-- 5. What data files and tablespaces are currently part of your database?
select tablespace_name, file_name
from dba_data_files;

-- 6. Use the v$logfile view to determine the location and name of your redo
log files.
select member from v$logfile;

-- 7. Use the v$archived_log view to determine the number of, location and name
of your archived redo log files, if any.
select name from v$archived_log;

-- 8. What mode is your database running in, NOARCHIVELOG or ARCHIVELOG?
select log_mode from v$database;

-- 9. How many control files do you have?  What are the location and name(s) of
the file(s)?
select name from v$controlfile;

-- 10. What is the name of the temporary tablespace assigned to your username?
select username, temporary_tablespace
from dba_users;
```

-- 11. How many temporary tablespaces have been created in your database?

```
select tablespace_name, contents
from dba_tablespaces
where contents = 'TEMPORARY';
```

-- 12. Query the v\$parameter view to determine the value of the db_cache_size parameter.

```
select name, value
from v$parameter
where name = 'db_cache_size';
```

-- 13. What users have SYSDBA privilege in your database?

```
select * from v$pwfile_users
where sysdba = 'TRUE';
```

--Lesson 3. Architecture / Processes Workshop

```
-- SkillBuilders, 2004

--
-- Workshop:  Architecture / Processes
--
-- Hands-on Exercises
--
-- 1. Connect to your instance with your DBA-level user.
connect system/dave@angela

-- 2. What background processes are active on your server?
--     Use both an operating system command (if using Unix or Linux)
--     and a SQL statement to list active processes.

!ps -aef | grep ora_

SELECT name, description
FROM v$bgprocess
WHERE paddr <> '00';

-- 3. What is the maximum number of archiver processes that can be started on
your instance?

show parameter log_archive_max_processes

-- Written Exercises

-- 1. What process will write dirty blocks back to a database file when a
checkpoint occurs?
--     Answer: The database writer, DBWR.

-- 2. What process will parse and execute SQL statements on behalf of a user?
--     Answer: A dedicated or shared server process.

-- 3. What process will, when a COMMIT is issued, write the redo log buffer
contents to the redo log file?
--     Answer: The log writer process, LGWR.
```

--Lesson 4. Architecture / Memory areas Workshop

```
-- SkillBuilders, 2004

--
-- Workshop: Architecture / Memory areas

-- 1. Q. What are the major memory structures in the SGA?
--     A. Buffer Cache, Redo buffer, shared pool (library cache and dictionary
--        cache), Large pool and Java pool. There is the potential for Keep
--        pool,
--        Recycle pool and sub-caches.

-- 2. Q. What actions cause the all or a portion of the Library Cache to be
--        flushed?
--     A. Altering a referenced object, ALTER SYSTEM FLUSH SHARED POOL
--        and database shutdown.

-- 3. Connect to your Oracle instance with the SYSTEM user.
connect system/dave@angela

-- 4. What is the size of the SGA in your instance?
select * from v$sga;
show parameter sga

-- 5. What is the size of your:
--     shared pool
show parameter shared_pool_size
--     java pool
show parameter java_pool
--     large pool
show parameter large_pool
--     keep pool
show parameter db_keep
--     recycle pool
show parameter db_recycle

-- 6. Are any sub-caches defined in your instance?
show parameter db_
```

--Lesson 5. Administrative Users Workshop

```
-- SkillBuilders, 2004

-- Oracle9i Database Administration course
-- Workshop Solution: Administrative Users

-- 1. Login to Oracle with the SYS user.

connect sys/dave as sysdba
-- Connected.

-- 2. Login to Oracle with SYSTEM user with the SYSDBA privilege.

connect system/dave as sysdba
-- Connected.

-- 3. Execute the SHOW USER command.  What user is returned?

show user
-- USER is "SYS"

-- 4. Login to Oracle with a local operating system connection.
--   If problems occur, check which OS users are part of the ORA_DBA OS group.

connect / as sysdba
-- Connected.

-- 5. Determine which users have been granted SYSDBA.

select * from v$pwfile_users;

-- USERNAME                SYSDB SYSOP
-- -----
-- SYS                       TRUE  TRUE
-- GARY                       TRUE  FALSE
-- SYSTEM                     TRUE  FALSE

-- 6. Create a new user.  Use your first name as the user name and password.

create user dave identified by dave;

-- User created.

-- 7. Grant SYSDBA to the new user.
```

```
grant sysdba to dave
/

-- Grant succeeded.

-- 8. Attempt to connect to the new user without the SYSDBA privilege.  What
happens?
connect dave/dave

-- The following error is returned because without SYSDBA because the user does
not have
-- the privilege to connect to the database (i.e. CREATE SESSION privilege):
-- ERROR: ORA-01045: user DAVE lacks CREATE SESSION privilege; logon denied

-- 9. Attempt to connect to the new user with the SYSDBA privilege.  What
happens?
connect dave/dave as sysdba
-- Connected.
-- This works because SYSDBA has the privilege to connect (i.e. CREATE SESSION
privilege).

-- 10. Grant the DBA role to the new user.
grant dba to dave;

-- Grant succeeded.

-- 11. Connect to the database with the new user - without SYSDBA
SQL> connect dave/dave
Connected.

SQL>
```

--Lesson 6. Startup Basics Workshop

-- SkillBuilders Oracle9i Database Administration Course

-- Workshop: Startup Basics

=====
===

-- 1. Setup SQL*Plus In Windows.

-- Follow the instructions listed in the course book to add the NOLOG
parameter

-- to a desktop SQL*Plus Windows icon.

-- 2. If your database instance is started, use SQL*Plus to shut it down,

-- then exit SQL*Plus. Use the new administrative user created in the

-- previous workshop.

CONNECT dave/dave AS SYSDBA

SHUTDOWN

EXIT

-- 3. Restart the database instance. Use all defaults on the STARTUP command.

CONNECT dave/dave AS SYSDBA

STARTUP

--Lesson 6. STARTUP MOUNT Workshop

```
-- SkillBuilders, 2004

-- Oracle9i Database Administration Course
-- Workshop:  STARTUP MOUNT

=====
PART 1
=====

-- 1. Connect with your new admin user and shutdown your database instance.

connect dave/dave as sysdba
-- Connected.
-- SQL> shutdown
-- Database closed.
-- Database dismounted.
-- ORACLE instance shut down.
-- SQL>

-- 2. Start it up in NOMOUNT state.

startup nomount
-- ORACLE instance started.

-- Total System Global Area  126950220 bytes
-- Fixed Size                  453452 bytes
-- Variable Size               96468992 bytes
-- Database Buffers            29360128 bytes
-- Redo Buffers                 667648 bytes
-- SQL>

-- 3. Attempt to connect without SYSDBA privilege.  What happens and why?
--    Connect with SYSDBA privilege.

CONNECT dave/dave
-- You get an error: "ORA-01033: ORACLE initialization or shutdown in
progress".
-- Only SYSDBA users can connect in this state.

connect dave/dave as sysdba
-- Connected.
-- SQL>

-- 4. Query v$instance and v$sga.

select instance_name from v$instance;
```

```
-- INSTANCE_NAME
-- -----
-- dave

SELECT * FROM v$sga;

-- NAME                                VALUE
-- -----
-- Fixed Size                          453452
-- Variable Size                        96468992
-- Database Buffers                    29360128
-- Redo Buffers                        667648

-- 5. Determine the database blocksize by querying v$parameter
-- WHERE NAME = 'db_block_size'. Note that the string "db_block_size"
-- must be in lower case.

select value
from v$parameter
where name = 'db_block_size';

-- VALUE
-- -----
-- 8192

-- 6. Attempt to query v$database and v$controlfile. What happens?

SELECT * FROM v$database;
-- You get an error: "ORA-01507: database not mounted" because the database is
-- in NOMOUNT mode and thus the control file is not open. V$DATABASE requires
-- information from the control file.

SELECT * FROM v$controlfile;

-- no rows selected

-- This displays "no rows selected" since in MOUNT mode the control file(s)
-- are not yet open.

=====
- PART 2
=====

-- 1. Mount the database.
alter database mount;
```

```
-- 2.
SELECT name FROM v$database;
SELECT instance_name, status FROM v$instance;
SELECT * FROM v$controlfile;
SELECT file#, status FROM v$datafile;
-- They all work because we are in MOUNT mode and the control file(s) are open.
-- The control file(s) have the names of all the datafiles.
-- The status shown in v$instance is MOUNT.

-- 3. Use ALTER to OPEN the database for normal operation.
ALTER DATABASE OPEN;

-- 4. Query v$instance. What is the database status show in v$instance.
SELECT status FROM v$instance;
-- The status shown in v$instance is OPEN.
```

--Lesson 6. Shutdown Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop:  Shutdown

-- 1. Start two SQL*Plus sessions.  Connect AS SYSDBA in one of the sessions.

-- 2. In the SYSDBA session, enter the SHUTDOWN command with no parameters.
--    What happens?

--    The session attempting to shutdown the database hangs.

-- 3. Unblock the hanging session.

-- In the 2nd session, enter exit or disc (disconnect)
SQL> disc
Disconnected from Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

-- The SYSDBA session continues with shutdown:
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.

-- 4. Restart the database.
startup

-- 5. Logon in the second session and enter the following commands:
connect dave/dave
create table t (c1 number);
insert into t values(1);

-- 6. In the SYSDBA session, Shutdown the database with the transactional
clause
--    What happens?
shutdown transactional

-- The shutdown is blocked until the second session commits.

-- 7. Unblock the shutdown.

-- In the second session, enter commit:
commit;
```

--Lesson 7. Server Parameters Files Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop Solution:  Server Parameter Files

-- 1. Check if your current instance is using a server parameter file.
--    If it is, proceed to step 4.  If not, proceed to step 2.

select value from v$parameter
where name = 'spfile';

-- If the value is NULL, you're NOT using an SPFILE.
select count(*) from v$spparameter
where value is not null;

-- If the count returned is 0 then you are using a text file.  If the
-- count returned is greater than 0 then you are using a spfile.

-- 2. Create a SPFILE from the text parameter file.

CREATE spfile FROM pfile;

-- 3. Restart your instance with the new spfile.
SHUTDOWN IMMEDIATE;
STARTUP;

-- 4. Reduce the size of the shared pool by 4 Mb.  Make this change persistent.
SHOW PARAMETER shared_pool_size;

-- NAME                                TYPE                                VALUE
-- -----
-- shared_pool_size                      big integer 16777216

ALTER SYSTEM SET shared_pool_size = 12000000 SCOPE=BOTH;
SHOW PARAMETER shared_pool_size;

-- NAME                                TYPE                                VALUE
-- -----
-- shared_pool_size                      big integer 12582912

-- 5. Restart the database with the SPFILE.  Check that the change made in step
--    4 is still in effect.
SHUTDOWN IMMEDIATE;
STARTUP;
SHOW PARAMETER shared_pool_size;
```

```

-- NAME                                TYPE                                VALUE
-- -----
-- shared_pool_size                    big integer 12582912

-- 6. Create a text version of the server parameter file.  View the contents.

CREATE pfile FROM spfile;

-- 7. Restart your instance with the text parameter file.
--     NOTE: Your path and file name may differ.

SHUTDOWN IMMEDIATE;
STARTUP pfile='c:\oracle\ora9i\database\initprod.ora'

-- 8. What is the current value of of your shared_pool_size parameter?
SHOW PARAMETER shared_pool_size;

-- NAME                                TYPE                                VALUE
-- -----
-- shared_pool_size                    big integer 12582912

-- 9.  Shutdown your database.
connect dave/dave as sysdba
shutdown;

-- 10. Remove your SPFILE by adding a ".x" to the end of its filename.

-- 11. Startup your database without any parameter.  Is the SPFILE being used?
connect dave/dave as sysdba
startup;
select value from v$parameter where name = 'spfile';
-- Note that the value now displays as null, indicating that the SPFILE is not
being used.

-- 12. Shutdown, then rename your SPFILE by removing the ".x" at the end for
the filename.
connect dave/dave as sysdba
Shutdown;
--Rename your SPFILE appropriately.

--13. Startup your database without any parameter.  Is the SPFILE being used?
connect dave/dave as sysdba
startup;
select value from v$parameter where name = 'spfile';
-- Note that the value now displays returns the filename of your SPFILE,
indicating that the SPFILE is being used.

```

--Lesson 8. User Basics Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop:  User Basics

-- 1. Create an Oracle user account with the following settings :
CREATE USER geoff2 IDENTIFIED BY geoff
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

-- 2. Query the Oracle data dictionary to ensure that this user has been
created correctly.
SELECT * FROM dba_users WHERE username = 'GEOFF2';

-- 3. Attempt to make a connection to the database using the user name and
password specified.
CONNECT geoff2/geoff

-- 4. Any ideas as to why it doesn't work? Fix the problem and re-connect.
--   The connect above fails because GEOFF2 lacks CREATE SESSION privilege.
CONNECT geoff/geoff
GRANT CREATE SESSION to geoff2;
CONNECT geoff2/geoff

-- The connect now succeeds.

-- 5. Change the user's password to yourname2 e.g. geoff2 .
ALTER USER geoff2 IDENTIFIED BY geoff2;

-- 6. Query the user_users data dictionary view while logged into the new user.
CONNECT geoff2/geoff2
SELECT * FROM user_users;

-- 7. Start a second session with user system then:
SELECT sid, serial#, username FROM v$session;
-- Assuming the original session has a sid of 7 and a serial# of 49:
ALTER SYSTEM KILL SESSION '7, 49';

-- 8. Now go to your original SQL*Plus session:
select user from dual;
-- select user from dual
-- *
-- ERROR at line 1:
-- ORA-00028: your session has been killed
```

-- Clearly the original session has been terminated.

--Lesson 8. User Quota & Account Status Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop:  User Quotas and Account status

-- 1.Create user your-name3 (e.g. geoff3) with a password of your-name3 (e.g.
geoff3), set its default tablespace to USERS and its temporary tablespace to
TEMP, and give the user a quota of 64K on the USERS tablespace.

connect system/dave

CREATE USER geoff3 IDENTIFIED BY geoff3
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA 64k ON users;

-- 2. Grant the necessary privileges to allow the user to connect and create
tables
GRANT CREATE SESSION, CREATE TABLE TO geoff3;

-- 3. Login to the new user in a SECOND SQL*Plus session and create this test
table:
connect geoff3/geoff3
CREATE TABLE test (dummy1 CHAR(2000));

-- 4. Determine how much space the test table is consuming.
select bytes from user_segments
where segment_name = 'TEST'
/

-- 5. Insert rows until a quota error is received:
INSERT INTO test VALUES ('test');
insert into test select * from test;
/
/
/
/

-- 6. In the SYSTEM session, set the users quota to unlimited:

ALTER USER geoff3 quota unlimited ON users;

-- 7 Retry an insert:
```

```
INSERT INTO test select * from test;

-- 8. Again determine how much space the test table is consuming.
select bytes from user_segments
where segment_name = 'TEST'
/

-- 9. Display all usernames and user account statuses.
select username, account_status
from dba_users
/

-- 10. Secure one of the open accounts.
alter user demo password expire account lock;
```

--Lesson 9. User Passwords Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop:  User Passwords

connect sys/password as sysdba

@verify_password
-- Note that this will create the following function:
CREATE OR REPLACE FUNCTION verify_function
  (username varchar2,
   password varchar2,
   old_password varchar2)
  RETURN boolean IS

BEGIN

  -- Check if the password is same as the username
  IF LOWER(password) = LOWER(username) THEN
    raise_application_error(-20001, 'Password same as or similar to user');
  END IF;

  -- Check for the minimum length of the password
  IF length(password) < 4 THEN
    raise_application_error(-20002, 'Password length less than 4');
  END IF;

  -- Check if the password is too simple. A dictionary of words may be
  -- maintained and a check may be made so as not to allow the words
  -- that are too simple for the password.
  IF LOWER(password) IN ('welcome', 'database', 'account', 'user', 'password',
'oracle', 'computer', 'abcd') THEN
    raise_application_error(-20002, 'Password too simple');
  END IF;

  -- Everything is fine; return TRUE ;
  RETURN(TRUE);
END;
/

create profile pwd
limit
password_verify_function verify_function;

alter user dave profile pwd;

connect dave/dave
```

```
-- Now user the SQL*Plus PASSWORD command to change the password as  
-- indicated in the lab.
```

```
-- Note that if you try the alter user command:  
alter user dave identified by dave;  
-- that a misleading error message is returned :  
-- ORA-28221: REPLACE not specified  
-- There seems to be a problem in Oracle if a user tries to use  
-- the ALTER USER command to change their own password when the  
-- password_verify_function is in use.
```

--Lesson 10. Auditing Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop:  Auditing

-- 1. Enable database auditing to the SYS.AUD$ table with the AUDIT_TRAIL
parameter.
connect system/dave as sysdba
-- Connected.

alter system
set audit_trail = db
scope =spfile;

-- System altered.

startup force
-- ORACLE instance started.

-- 2. Audit all connections to the database.

audit session;

-- Audit succeeded.

-- 3. Test by connecting to the database and querying the appropriate view.

connect dave/dave
-- Connected.

connect system/dave
-- Connected.

@aud_login


```

USERNAME	TERMINAL	result	time	logoff
DAVE	DAVELAPTOP	Connected	10/14/03 17:39	10/14/03 17:39
SYSTEM	DAVELAPTOP	Connected	10/14/03 17:39	

```

-- 4. Disable auditing.
noaudit session;

-- Noaudit succeeded.
```

```
alter system
set audit_trail=false
scope=spfile;

-- System altered.

connect system/dave as sysdba
-- Connected.
startup force
```

--Lesson 11. System Privileges Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop:  System Privileges

-- 1. Logon to SQL*Plus with your DBA user.
connect dave/dave

-- 2. List the system privileges currently in effect for your session.
select * from session_privs;

-- 3. Create a new user called TABLEMAN

create user tableman identified by tableman
default tablespace users
quota unlimited on users;

-- User created.

-- 4. Give the user the ability to logon and create tables
grant create session, create table to tableman;

-- Grant succeeded.

-- 5. Logon to the new user.
connect tableman/tableman
-- Connected.

-- 6. Create a simple test table in the TABLEMAN schema.
create table t (c1 number);

-- Table created.

-- 7. While connected to the TABLEMAN schema, try to create a simple table in
the SYSTEM schema.
--   What happens?  How would you permit this?

CREATE TABLE SYSTEM.t1 (c1 NUMBER);
-- CREATE TABLE SYSTEM.t1 (c1 NUMBER)
-- *
-- ERROR at line 1:
-- ORA-01031: insufficient privileges
```

```
-- An error occurred because user TABLEMAN does not have CREATE ANY TABLE
privilege.
-- To correct:

-- SQL> connect dave/dave@class2
-- Connected.
-- SQL> revoke create table from tableman;
--
-- Revoke succeeded.
--
-- SQL> grant create any table to tableman;
--
-- Grant succeeded.
-- SQL> connect tableman/tableman@class2
-- Connected.
-- SQL> CREATE TABLE SYSTEM.t1 (c1 NUMBER);
--
-- Table created.
```

--Lesson 11. Object Privileges Workshop

```
-- SkillBuilders, 2004

-- Oracle9i Administration Course, Workshop Solution: Object Privileges

-- 1. Logon to the TABLEMAN user you created in the previous workshop in this
lesson.
connect tableman/tableman@class2

-- 2. Grant all privileges on your "T" table to user SYSTEM (or the DBA user
you created for this class).
grant all on t to dave;

-- 3. Grant select and insert privileges on your "T" table to user SCOTT.
grant select, insert on t to scott;

-- 4. Query the data dictionary to determine all recipients of object
privileges on your table.
select distinct grantee from user_tab_privs_made
where table_name = 'T';

-- 5. Connect to user SCOTT and query the data dictionary to view all table
privileges on objects owned by TABLEMAN.
connect scott/tiger@class2
select table_name, privilege
from user_tab_privs
where owner = 'TABLEMAN';

-- TABLE_NAME                PRIVILEGE
-- -----
-- T                            INSERT
-- T                            SELECT

-- 6. Re-connect to TABLEMAN and revoke all privileges on your "T" table.
connect tableman/tableman@class2
-- Connected.
revoke all on t from dave, scott;

-- Revoke succeeded.
```

--Lesson 11. Roles Workshop

```
-- SkillBuilders Oracle9i Database Administration Course
-- Workshop: Roles

-- 1. Create a role called 'manager' and assign that role the ability
to be able to create tables
-- and synonyms. Create another role called 'salesperson' which is
protected by a password.
-- The 'salesperson' role should be given the ability to be able to
make connections to the database.
```

```
SQL>CREATE ROLE manager;
```

```
Role created.
```

```
SQL>
```

```
SQL> grant create table to manager;
```

```
Grant succeeded.
```

```
SQL>
```

```
SQL> create role salesperson identified by dave;
```

```
Role created.
```

```
SQL> grant create session to salesperson;
```

```
Grant succeeded.
```

```
-- 2. Give the 'manager' role the ability to be able to insert, update
rows on the U1.delegates
```

```
-- table whereas the 'salesperson' role can only query them.
```

```
SQL> grant insert, update on ul.delegates to manager;
```

```
Grant succeeded.
```

```
SQL>
```

```
SQL> grant select on ul.delegates to salesperson;
```

```
-- 3. Assign the 'salesperson' role to the 'manager' role since the
manager will require all the
```

```
-- privileges given to the 'salesperson' role.
```

```
grant salesperson to manager;
```

```
-- 4. Give both roles to the U3 Oracle user (created in the last  
exercise).
```

```
grant manager to u3;
```

```
-- 5. Connect as the U3 user and determine which roles are present.  
Access the U1.delegates table  
-- and determine what actions (insert, update, delete) are allowed.
```

```
SQL> connect u3/u3
```

```
Connected.
```

```
SQL>
```

```
SQL> select d_lastname from u1.delegates;
```

```
D_LASTNAME
```

```
-----
```

```
test
```

```
SQL>
```

```
SQL> insert into u1.delegates
```

```
  2 values( 1, 'test', 'test', 'test', 'M', sysdate);
```

```
1 row created.
```

```
SQL>
```

```
SQL> update u1.delegates
```

```
  2 set d_date_last_contacted = sysdate - 1;
```

```
2 rows updated.
```

```
-- 6. Enable only the 'salesperson' role and again determine which  
actions are allowed in the U3 user.
```

```
SQL> set role salesperson identified by dave;
```

```
Role set.
```

```
SQL> select d_lastname from u1.delegates;
```

```
D_LASTNAME
```

```
-----
```

```
test
```

test

```
SQL> insert into u1.delegates
  2  values( 1, 'test', 'test', 'test', 'M', sysdate);
insert into u1.delegates
```

*

```
ERROR at line 1:
ORA-01031: insufficient privileges
```

```
SQL> update u1.delegates
  2  set  d_date_last_contacted = sysdate - 1;
update u1.delegates
```

*

```
ERROR at line 1:
ORA-01031: insufficient privileges
```

--Lesson 12. Tablespaces Basics Workshop

```
-- SkillBuilders, 2004

--
-- Oracle9i Administration
--
-- Workshop Solution - Tablespace Basics
--

-- 1. What tablespaces currently exist on your database and
--    what type of data do they contain?

select tablespace_name, status, contents
from dba_tablespaces;

-- 2. What database files exist on your database and what
--    tablespace do they belong to?

select tablespace_name, file_name
from dba_data_files
order by 1;

-- 3. What type of tablespace is your system tablespace,
--    dictionary managed or locally managed?
select extent_management
from dba_tablespaces
where tablespace_name = 'SYSTEM';

-- 4. If the system tablespace is an LMT, what type of tablespace will
--    all other tablespaces have to be?
--    A. LMT
```

--Lesson 12. Tablespaces Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop: Tablespaces

-- 1.Enable OMF (if OMF) is not enabled and create a OMF managed tablespace
called ts1 with the following attributes:
-- Datafile size 10m
-- Automatic Segment Space Management = AUTO
-- Locally managed with Uniform extent size = 1M
-- Autoextend off.
SQL> connect system/dave
Connected.
SQL> show parameter db_create_file_dest
```

NAME	TYPE	VALUE
db_create_file_dest	string	

```
1 alter system
2 set db_create_file_dest='C:\oracle\oradata\dave\'
3* scope=both
SQL> /
```

System altered.

```
SQL> create tablespace ts1
2 datafile size 10m autoextend off
3 extent management local uniform size 1m
4 segment space management auto
5 /
```

Tablespace created.

```
-- 2.Query the dba_data_files view to determine the name of the physical
datafile created.
```

```
SQL> select file_name, bytes
2 from dba_data_files
3 where tablespace_name = 'TS1'
4 /
```

FILE_NAME

BYTES

```
C:\ORACLE\ORADATA\DAVE\O1_MF_TS1_ZRTXK800_.DBF
10485760
```

```
-- 3.Create a test table called "T" in tablespace TS1.
```

```
SQL> create table t
2 (c1 number)
3 tablespace ts1;
```

```
Table created.
```

```
-- 4.Query the DBA_SEGMENTS view to determine the amount of space used by table T.
```

```
SQL> select bytes, blocks
2 from dba_segments
3 where segment_name = 'T'
4 and owner = user;
```

```

      BYTES      BLOCKS
-----
1048576         128
```

```
-- 5.Run the FREE_SPACE.SQL script and review the report to determine if some tablespaces should have space added.
```

```
SQL> @free_space
```

Tablespace Name	KBytes	Used	Free	% Used	Largest
CWMLITE	10,240	9,600	640	93.8	384
DRSYS	10,240	9,920	320	96.9	320
EXAMPLE	168,960	167,552	1,408	99.2	960
INDX	10,240	64	10,176	.6	10,176
ODM	10,240	9,536	704	93.1	704
SYSTEM	573,440	456,000	117,440	79.5	115,648
TOOLS	10,240	6,208	4,032	60.6	4,032
TS1	10,240	2,048	8,192	20.0	8,192
TS2	10,240	1,024	9,216	10.0	9,216
UNDOTBS1	419,840	20,800	399,040	5.0	212,992
USERS	512,000	384	511,616	.1	511,616
XDB	39,040	38,976	64	99.8	64
sum	1,784,960	722,112	1,062,848		

```
12 rows selected.
```

```
-- 6.Add 1M of disk space to one of the tablespaces.
```

```

1 alter tablespace example
2* add datafile size 1m
SQL> /

```

Tablespace altered.

-- 7.Run the FREE_SPACE.SQL script and review the report to determine if some tablespaces should have space added.

```
SQL> @free_space
```

Tablespace Name	KBytes	Used	Free	% Used	Largest
CWMLITE	10,240	9,600	640	93.8	384
DRSYS	10,240	9,920	320	96.9	320
EXAMPLE	169,960	167,552	2,408	98.6	1000
INDX	10,240	64	10,176	.6	10,176
ODM	10,240	9,536	704	93.1	704
SYSTEM	573,440	456,000	117,440	79.5	115,648
TOOLS	10,240	6,208	4,032	60.6	4,032
TS1	10,240	2,048	8,192	20.0	8,192
TS2	10,240	1,024	9,216	10.0	9,216
UNDOTBS1	419,840	20,800	399,040	5.0	212,992
USERS	512,000	384	511,616	.1	511,616
XDB	39,040	38,976	64	99.8	64
sum	1,784,960	722,112	1,062,848		

12 rows selected.

--Lesson 13. Table Create Workshop

```
-- SkillBuilders Oracle9i Database Administration Course

-- Workshop - Table Create Solution
-- 1. Use your username (e.g. "dave") to create a table (called "T") that
contains the following columns:
--   C1 NUMBER
--   C2 DATE
--   C3 TIMESTAMP
--   Place it in the TS1 tablespace created in the previous lesson.  If this
tablespace is not available, place the table in the USERS tablespace.
```

```
SQL> connect system/dave
Connected.
```

```
SQL> alter user dave quota unlimited on ts1;
```

User altered.

```
SQL> connect dave/dave
```

```
SQL> create table t
  2  (c1 number,
  3  c2 date,
  4  c3 timestamp)
  5  tablespace ts1;
```

Table created.

```
-- 2. Insert a row into the new table.  Use SYSDATE and SYSTIMESTAMP for the
--   DATE and TIMESTAMP columns.
```

```
SQL> insert into t
  2  values( 1, sysdate, systimestamp);
```

1 row created.

```
-- 3. Select the row from your table.
```

```
SQL> select * from t;
```

```
      C1 C2
-----
C3
-----
      1 15-OCT-03
15-OCT-03 04.12.15.730000 PM
```

--Lesson 13. CTAS Workshop

```
--  
-- Oracle9i Database Administration Workshop Solution - CTAS  
  
-- 1. Use the CREATE TABLE AS statement to create a new table based  
--    on the DBA_OBJECTS view. Place your new table in the USERS tablespace.  
  
create table t tablespace users as select * from dba_objects;
```

--Lesson 13. DBMS_STATS Workshop

```
--
-- Oracle9i Database Administration
-- DBMS_STATS Workshop Solution

-- 1. Run the supplied scripts ddl.sql and insert.sql to create tables and
-- indexes needed for this workshop.
@ddl
@insert

-- 2. Run the following command to delete all table and
-- index statistics in your schema:
EXECUTE DBMS_STATS.DELETE_SCHEMA_STATS (USER);

-- 3. Run the following command:
EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS (USER);

-- 4. Did the command in question 2 create index statistics?
-- Run this command to find out:
SELECT index_name
FROM user_indexes
WHERE last_analyzed IS NULL;

-- No, several rows were returned revealing that index stats were not
-- collected.
-- 5. Run DBMS_STATS.GATHER_SCHEMA_STATS to collect stats for all of your
-- tables and indexes. Limit histograms to just the indexed columns.

exec dbms_stats.gather_schema_stats(ownname=> USER, -
cascade=>TRUE , method_opt=>'FOR COLUMNS AUTO')

6. Verify the creation of the statistics by running the following queries:

SELECT table_name
FROM user_tables
WHERE last_analyzed IS NULL;

SELECT index_name
FROM user_indexes
WHERE last_analyzed IS NULL;

-- There are a few LOB indexes w/o statistics, but this is normal.
-- All objects have current statistics.
```

--Lesson 13. Table Compression Workshop

```
-- Workshop - Table Compression Solution

-- 1. Run the supplied script TABLE_COMPRESSION_PREP.SQL. This script creates 2
tables,
-- T1 and T2. Both tables are identical except that T2 was created with the
COMPRESS option.

@TABLE_COMPRESSION_PREP.SQL

-- 2. Query the data dictionary to view the difference in space used between
tables T1 and T2.

SELECT bytes, blocks FROM user_segments
WHERE segment_name = 'T1';

--      BYTES      BLOCKS
-- -----
--      4194304      1024

SELECT bytes, blocks FROM user_segments
WHERE segment_name = 'T2';

--      BYTES      BLOCKS
-- -----
--      2097152      512

-- 3. Execute the following SQL*Plus command to turn on automatic tracing.
SET AUTOTRACE TRACEONLY;

-- 4. Execute the following queries and note the differences in I/O counts for
the two queries.

SELECT object_name FROM t1
WHERE owner = 'DAVE';

-- Statistics
-- -----
--          0 recursive calls
--          0 db block gets
--         831 consistent gets
--         818 physical reads
--           0 redo size
--        1451 bytes sent via SQL*Net to client
```

```
--          532 bytes received via SQL*Net from client
--          5  SQL*Net roundtrips to/from client
--          0  sorts (memory)
--          0  sorts (disk)
--          48  rows processed
```

```
SELECT object_name FROM t2
WHERE owner = 'DAVE';
```

```
-- Statistics
```

```
-- -----
--          0  recursive calls
--          0  db block gets
--         434  consistent gets
--         426  physical reads
--          0  redo size
--       1451  bytes sent via SQL*Net to client
--          532  bytes received via SQL*Net from client
--          5  SQL*Net roundtrips to/from client
--          0  sorts (memory)
--          0  sorts (disk)
--          48  rows processed
```

```
-- 5. Execute the supplied script TABLE_COMPRESSION_RESET.SQL.
```

```
--@TABLE_COMPRESSION_RESET;
```

--Lesson 14. Indexes Workshop

-- Workshop: Indexes

-- 1. Review and Run the supplied script SETUP_INDEX_WORKSHOP.SQL.
-- Note the elapsed time to run the PL/SQL program.

@setup_index_workshop

SQL> -- Run a PL/SQL program to repeatedly fetch from T.

SQL> -- Display the elapsed time.

SQL>

SQL> set serveroutput on

SQL>

SQL> declare

2 v1 number;

3 l_start number := dbms_utility.get_time;

4 begin

5 for x in 1..1000 loop

6 begin

7 select c2

8 into v1

9 from t

10 where c1 = x;

11 exception

12 when no_data_found then null;

13 end;

14 end loop;

15

16 dbms_output.put_line(

17 round((dbms_utility.get_time - l_start) / 100,2) || ' seconds');

18 end;

19 /

4.99 seconds

PL/SQL procedure successfully completed.

-- 2. Create an index on table "T" to reduce the elapsed time.

SQL> create index t_idx on t (c1,c2) nologging compute statistics;

Index created.

-- 3. Rerun the PL/SQL program. Insure that the elapsed time has been reduced.

SQL> declare

```
2  v1 number;
3  l_start number := dbms_utility.get_time;
4  begin
5  for x in 1..1000 loop
6  begin
7  select c2
8  into v1
9  from t
10 where c1 = x;
11 exception
12 when no_data_found then null;
13 end;
14 end loop;
15
16 dbms_output.put_line(
17   round( (dbms_utility.get_time - l_start) / 100,2) || ' seconds');
18 end;
19 /
.09 seconds
```

PL/SQL procedure successfully completed.

--Lesson 14. Composite Index Workshop

-- Workshop Solution - Composite Index

-- 1. Run the supplied script "bigtable.sql". Describe the BIG table to become familiar with the makeup of the table.

```
@bigtable
```

```
desc big
```

-- 2. Use AUTOTRACE to trace the following query

```
set autotrace traceonly
```

```
select * from big
where object_type = 'TABLE'
and status = 'INVALID';
```

```
set autotrace off
```

-- 3. What is the access path used?

-- A. Full Table Scan

-- 4. Create an index on the two columns used in the query. Which column should be first?

-- Determine the selectivity of the columns:

```
select count(distinct object_type), count(distinct status) from big;
```

-- A. Since status has less selectivity, I'll use status as the leading column.

```
create index big2 on big (status, object_type) compress nologging compute
statistics;
```

-- 5. Re-trace the query? What was the access path used?

```
set autotrace traceonly
```

```
select * from big
where object_type = 'TABLE'
and status = 'INVALID';
```

```
set autotrace off
```

-- A. An Index Range Scan was used.

--Lesson 15. Constraints Workshop

```
--
-- Oracle9i Database Administration Workshop Solution:  Constraints
--
--
-- 1. Create a table called "T" that contains the following columns and
--     constraints (see bottom of page).  Provide meaningful names for all
--     constraints.
-- 2. Test the constraints by inserting one or more rows into your table.
--     Use values that will fail constraint validation.
--     Successfully insert at least one row into the table.
--
-- col name  datatype      constraint
-----
-- c1        number        not null
-- c2        number        not null & greater than 0
-- c3        date          not null & after December 31, 1999.
-- c4        char(1)       'Y', 'N'
-- c5        date          greater than or equal to c3
-- c1 & c2 are primary key

-- Cleanup
drop table t;

create table t
(
  c1 number,
  c2 number constraint t_c2_value check (c2 > 0),
  constraint t_pk primary key (c1, c2),
  c3 date
      constraint t_c3_value check (c3 >= to_date('12/31/1999','mm/dd/yyyy'))
      constraint t_c3_nn not null,
  c4 char(1)
      constraint t_c4_value check (c4 in ('Y', 'N') ),
  c5 date,
  constraint t_c5_value check (c5 >= c3)
);

insert into t
(c1, c2, c3, c4, c5)
values
(1, 10, sysdate, 'Y', sysdate+1);

-- Demo some constraint violations:

-- SQL> insert into t
```

```
-- 2 (c1, c2, c3, c4, c5)
-- 3 values
-- 4 (1, 10, sysdate, 'Y', sysdate-1);
-- insert into t
-- *
-- ERROR at line 1:
-- ORA-02290: check constraint (DAVE.T_C5_VALUE) violated

-- SQL> insert into t
-- 2 (c1, c2, c3, c4, c5)
-- 3 values
-- 4 (1, 10, sysdate, 'Y', sysdate+1);
--
-- 1 row created.

insert into t
(c1, c2, c3, c4, c5)
values
(1, 10, sysdate, 'Y', sysdate+1);
```

--Lesson 15. Constraint Exceptions Workshop

```
--
-- Oracle9i Database Administration Workshop Solution: Constraint Exceptions
--
--
-- 1. Run the supplied script setup_constraint_workshop.sql.
@setup_constraint_workshop

-- 2. Create a table called "T" based on the supplied table
-- "CONSTRAINT_WORKSHOP_T". Use a CREATE TABLE AS statement to
-- build your new table. Copy all rows from the supplied table
-- into your new table.

-- cleanup
drop table t;

create table t as select * from constraint_workshop_t;

-- 3. Add a primary key constraint to the "T" table on columns C1 and C2.
-- Use the EXCEPTIONS INTO clause to capture ROWIDs of rows that fail
-- validation.

-- create the exceptions table
@utlexcpt

alter table t add constraint t_pk primary key(c1,c2)
exceptions into exceptions;

-- 4. Delete all rows that fail validation.

delete from t where rowid in (select row_id from exceptions);

-- 5. Add the constraint.
alter table t add constraint t_pk primary key(c1,c2)
exceptions into exceptions;
```

--Lesson 15. Constraint Indexes Workshop

```
--
-- Oracle9i Database Administration Workshop Solution: Constraint Indexes
--

-- 1. Describe the supplied table CW_TAB (this table was created by
-- executing the supplied script setup_constraint_workshops.sql.
desc cw_tab

-- 2. Add a unique constraint to the C1 column.

alter table cw_tab add constraint c1_unique unique(c1);

-- 3. Add a unique constraint to the C1 and C2 columns (composite).

alter table cw_tab add constraint c2_unique unique(c1, c2);

-- 4. Execute the following query. What index did steps 2 and 3 create or
-- use for the constraints you added? Why?
select constraint_name, index_name
from user_constraints
where table_name = 'CW_TAB';

-- SQL> select constraint_name, index_name
-- 2 from user_constraints
-- 3 where table_name = 'CW_TAB';
--
-- CONSTRAINT_NAME          INDEX_NAME
-- -----
-- C1_UNIQUE                 CW_TAB_I
-- C2_UNIQUE                 CW_TAB_I

-- Oracle used the pre-built index "CW_TAB_I" because it was a suitable
-- index for both unique constraints, i.e. the leading edge of the index
-- matched the leading edge of the constraints...
```

--Lesson 16. Automatic Undo Management Workshop

```
-- Workshop - Automatic UNDO Management

-- 1. What are the current values of the following parameters for your
database?
--     UNDO_MANAGEMENT, UNDO_TABLESPACE, UNDO_RETENTION, UNDO_SUPPRESS_ERRORS.

show parameter UNDO_MANAGEMENT;
show parameter UNDO_TABLESPACE;
show parameter UNDO_RETENTION;
show parameter UNDO_SUPPRESS_ERRORS;

-- Or, the following command will show the values for all four parameters.

show parameter UNDO;

-- 2. Create a new tablespace called UNDO_DATA_TS with the intent that this
tablespace
--     will be used as your new UNDO tablespace. Use OMF if active.
--     Create a 1M datafile with AUTOEXTEND OFF.
CREATE UNDO TABLESPACE undo_data_ts
DATAFILE SIZE 1M AUTOEXTEND OFF;

-- 3. Make the tablespace UNDO_DATA_TS your new UNDO tablespace for your
database.

ALTER SYSTEM SET UNDO_TABLESPACE = undo_data_ts
COMMENT='SkillBuilders AUM workshop' SCOPE=BOTH;

show parameter undo_tablespace

-- 4. Execute the supplied script AUM1.SQL. This script will generate an error
--     related to undo.

@AUM1
-- ORA-30036: unable to extend segment by 16 in undo tablespace

-- 5. Query V$UNDOSTAT to determine the number of undo blocks consumed in the
--     last 60 minutes. Was there any attempted stealing of unexpired blocks?

select to_char(begin_time, 'hh:mi:ss' ) as time,
       undoblks, unxpstealcnt
from v$undostat
where begin_time > sysdate - 1/24
order by 1 desc;

-- Q. If you experienced any attempted stealing of unexpired blocks,
```

```
--      what would you do to correct this?
-- A. Increase UNDO tablespace size to correct this situation.

-- 6. Revert to the original undo tablespace.  Rerun supplied script AUM1.SQL.
--      Insure there are no errors.
alter system set undo_tablespace=undotbs
scope =both;

@auml
```

--Lesson 17. Archiving Workshop

-- Workshop Solution: Enabling Archiving

-- 1. Check the Archive mode of your database:

```
connect / as sysdba
archive log list;
```

```
-- Database log mode           No Archive Mode
-- Automatic archival         Disabled
```

-- 2. Check the value of the LOG_ARCHIVE_START command.

-- Though the output of the ARCHIVE LOG LIST command tells me that the
-- "Automatic archival" is off ("Disabled"), I can also check this with the
value of the

-- LOG_ARCHIVE_START command:

```
SQL> show parameter log_archive_start
```

NAME	TYPE	VALUE
log_archive_start	boolean	FALSE

-- 3. If the database is not in archive log mode, enable it. If the automatic
-- archival process is not started, start it.

```
alter system
set log_archive_start=true
scope=spfile;
```

-- System altered.

```
shutdown
```

```
-- Database closed.
-- ORACLE instance shut down.
```

```
startup mount
```

-- ORACLE instance started.

```
alter database archivelog;
```

-- Database altered.

```
alter database open;
```

--Lesson 17. Multiple Archive Logs Workshop

```
-- Workshop Solution: Multiplexing Archive Logs

-- 1. Determine the current status of your databases archive logs, with respect
--    to location and multiplexing.

show parameter log_archive_dest
col destination format a35
col dest_name format a25

select dest_name, status, binding, destination
from v$archive_dest;

-- 2. If your database is not multiplexing the archive redo log, multiplex the
log.
--    Specify two mandatory locations for the log.  If the computer you're
using
--    for these workshops has two disks available, place the archive logs on
separate
--    disks.

alter system
set log_archive_dest = ''
scope = both
/

-- System altered.

alter system
set log_archive_dest_1 = "location=c:\oracle\oradata\dave\archive\ mandatory"
scope = both
/

-- System altered.

alter system
set log_archive_dest_2 = "location=c:\oracle\oradata\dave\archive2\ mandatory"
scope = both
/

-- System altered.

-- 3. Query the V$ARCHIVE_DEST view to insure the configuration multiplexes the
--    archive log.

col destination format a35
```

```
col dest_name format a25
```

```
select dest_name, status, binding, destination
from v$archive_dest;
```

```
-- DEST_NAME          STATUS    BINDING  DESTINATION
-----
```

```
-- LOG_ARCHIVE_DEST_1      VALID    MANDATORY
c:\oracle\oradata\dave\archive\
-- LOG_ARCHIVE_DEST_2      VALID    MANDATORY
c:\oracle\oradata\dave\archive2\
-- LOG_ARCHIVE_DEST_3      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_4      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_5      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_6      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_7      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_8      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_9      INACTIVE OPTIONAL
-- LOG_ARCHIVE_DEST_10     INACTIVE OPTIONAL
```

```
-- 10 rows selected.
```

--Lesson 17. Add Redo Log Workshop

```
-- Workshop Solution: Redo Logs

-- 1. Use V$LOG and V$LOGFILE to determine the number and size of the
--     existing log files.

select group#, member
from v$logfile;

--      GROUP# MEMBER
-- -----
--      4 C:\ORACLE\ORADATA\DAVE\O1_MF_4_ZR620B00_.LOG
--      4 C:\ORACLE\ORADATA\DAVE\LOGS\O1_MF_4_ZR620O00_.LOG
--      5 C:\ORACLE\ORADATA\DAVE\O1_MF_5_ZR62PK00_.LOG
--      5 C:\ORACLE\ORADATA\DAVE\LOGS\O1_MF_5_ZR62PV00_.LOG

select group#, bytes, members
from v$log;

--      GROUP#      BYTES      MEMBERS
-- -----
--      4      104857600          2
--      5      104857600          2

-- 2. Add another multiplexed OMF log file to the database.

-- Is my database using OMF?
show parameter db_create_online

--NAME                                                    TYPE          VALUE
-----
--db_create_online_log_dest_1                          string        C:\ORACLE\ORADATA\DAVE\
--db_create_online_log_dest_2                          string        C:\ORACLE\ORADATA\DAVE\LOGS\
--db_create_online_log_dest_3                          string
--db_create_online_log_dest_4                          string
--db_create_online_log_dest_5                          string

-- Yes, my database is using OMF. Since this is true and the size of my
existing
-- logs is 100Mb, I can add another mutliplexed member easily:

alter database add logfile;

-- Database altered.
```

-- If the database OMF parameters were not set, I would execute the following commands:

```
-- SQL> alter system set
--   2 db_create_online_log_dest_1='C:\ORACLE\ORADATA\DAVE\'
--   3 scope=both;
```

-- System altered.

```
-- SQL> alter system set
--   2 db_create_online_log_dest_2='D:\ORACLE\ORADATA\DAVE\'
--   3* scope=both;
```

-- System altered.

--Lesson 18. Add Control File Workshop

```
-- Workshop Solution: Adding Control File

-- 1. Determine the control files currently in use.
select value
from v$parameter2
where name = 'control_files';

-- OR

select name
from v$controlfile;

-- 2. Add another control file to your database.
shutdown transactional

$copy C:\ORACLE\ORADATA\DAVE\CONTROL01.CTL C:\ORACLE\ORADATA\DAVE\CONTROL03.CTL

create pfile from spfile;

startup nomount

alter system
set control_files = 'C:\oracle\oradata\dave\CONTROL01.CTL' ,
                   'C:\oracle\oradata\dave\CONTROL02.CTL' ,
                   'C:\oracle\oradata\dave\CONTROL03.CTL'
scope = spfile;

shutdown

startup

-- 3. Check that the new control file is in use.
select value
from v$parameter2
where name = 'control_files';

-- OR

select name
from v$controlfile;
```

--Lesson 18. Optional Create Control File Workshop

```
-- Workshop Solution: Creating Control File

-- This is an optional workshop.  The instructor will assign only if time
permits.

-- 1. Execute the ALTER DATABASE command that creates the CREATE CONTROLFILE
command script.
alter database backup controlfile to trace;
-- Database altered.

show parameter user_dump_dest

select a.spid
from v$process a, v$session b
where a.addr = b.paddr
and b.audsid = userenv('sessionid');

-- SPID
-- -----
-- 1160

-- 2. Shutdown your database.
shutdown
--Database closed.
--Database dismounted.
--ORACLE instance shut down.

-- 3. Rename or move all control files to an alternate location.
-- Do not delete, lose or edit the control files (in case we need them for
recovery).

-- 4. Try to start your database.
-- The startup will fail because the control files cannot be found.
startup
-- ORACLE instance started.
--
-- Total System Global Area 126950220 bytes
-- Fixed Size 453452 bytes
-- Variable Size 96468992 bytes
-- Database Buffers 29360128 bytes
-- Redo Buffers 667648 bytes
-- ORA-00205: error in identifying controlfile, check alert log for more info

-- 5. Locate the error message in the alert file.
```

```
-- ** The alert log in the BDUMP directory contains: **

-- ORA-00202: controlfile: 'C:\oracle\oradata\dave\CONTROL01.CTL'
-- ORA-27041: unable to open file
-- OSD-04002: unable to open file
-- O/S-Error: (OS 2) The system cannot find the file specified.

-- Fri Oct 03 14:10:07 2003
-- ORA-205 signalled during: ALTER DATABASE  MOUNT...

-- 6. Recreate all your control files and startup your database by executing
the
-- script created in step 1.
@create_controlfile

-- 7. Shutdown and restart your database. The startup should succeed.
shutdown

-- Database closed.
-- Database dismounted.
-- ORACLE instance shut down.

startup
-- ORACLE instance started.

-- Total System Global Area 126950220 bytes
-- Fixed Size 453452 bytes
-- Variable Size 96468992 bytes
-- Database Buffers 29360128 bytes
-- Redo Buffers 667648 bytes
-- Database mounted.
-- Database opened.
--SQL>
```

--Lesson 19. RMAN Workshop

```
-- Workshop: RMAN

-- 1. Determine which datafiles are in need of backup.

-- first, start RMAN from the OS command prompt:
c:\oracleScripts\rman> rman target system/dave

Recovery Manager: Release 9.2.0.4.0 - Production

Copyright (c) 1995, 2002, Oracle Corporation. All rights reserved.

connected to target database (not started)

-- My database was not started, so I can start it from RMAN:
RMAN> startup

Oracle instance started
database mounted
database opened

Total System Global Area      210837176 bytes

Fixed Size                    454328 bytes
Variable Size                 134217728 bytes
Database Buffers              75497472 bytes
Redo Buffers                  667648 bytes

-- Now I can determine which datafiles are in need of backup:
RMAN> report need backup;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 2
Report of files with less than 2 redundant backups
File #bkps Name
-----
1      0      C:\ORACLE\ORADATA\DAVE\SYSTEM01.DBF
2      0      C:\ORACLE\ORADATA\DAVE\UNDOTBS01.DBF
3      0      C:\ORACLE\ORADATA\DAVE\CWMLITE01.DBF
4      0      C:\ORACLE\ORADATA\DAVE\DRSYS01.DBF
5      1      C:\ORACLE\ORADATA\DAVE\EXAMPLE01.DBF
6      0      C:\ORACLE\ORADATA\DAVE\INDX01.DBF
7      0      C:\ORACLE\ORADATA\DAVE\ODM01.DBF
8      0      C:\ORACLE\ORADATA\DAVE\TOOLS01.DBF
9      0      C:\ORACLE\ORADATA\DAVE\O1_MF_TS1_ZRTXK800_.DBF
10     0      C:\ORACLE\ORADATA\DAVE\XDB01.DBF
```

```
11 1 C:\ORACLE\ORADATA\DAVE\O1_MF_USERS_ZMNKR400_.DBF
12 0 C:\ORACLE\ORADATA\DAVE\TS2.DBF
13 1 C:\ORACLE\ORADATA\DAVE\O1_MF_EXAMPLE_ZRTXR700_.DBF
14 0 C:\ORACLE\ORADATA\DAVE\O1_MF_UNDO_DAT_ZRXRRD00_.DBF
```

RMAN>

```
-- 2. Configure your RMAN environment so that all backups include an automatic
--- backup of the control file.
```

RMAN> configure controlfile autobackup on;

old RMAN configuration parameters:

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

new RMAN configuration parameters:

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

new RMAN configuration parameters are successfully stored

```
-- 3. Backup one of the tablespaces associated with a datafile requiring
backup.
```

--

RMAN> backup full tablespace example;

Starting backup at 16-OCT-03

using channel ORA_DISK_1

using channel ORA_DISK_2

channel ORA_DISK_1: starting full datafile backupset

channel ORA_DISK_1: specifying datafile(s) in backupset

input datafile fno=00013

name=C:\ORACLE\ORADATA\DAVE\O1_MF_EXAMPLE_ZRTXR700_.DBF

channel ORA_DISK_1: starting piece 1 at 16-OCT-03

channel ORA_DISK_2: starting full datafile backupset

channel ORA_DISK_2: specifying datafile(s) in backupset

input datafile fno=00005 name=C:\ORACLE\ORADATA\DAVE\EXAMPLE01.DBF

channel ORA_DISK_2: starting piece 1 at 16-OCT-03

channel ORA_DISK_1: finished piece 1 at 16-OCT-03

piece handle=C:\ORACLE\BACKUPS\BACKUP_0PF425FS_1_1 comment=NONE

channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03

channel ORA_DISK_2: finished piece 1 at 16-OCT-03

piece handle=C:\ORACLE\BACKUPS\BACKUP_0QF425FS_1_1 comment=NONE

channel ORA_DISK_2: backup set complete, elapsed time: 00:00:18

Finished backup at 16-OCT-03

Starting Control File and SPFILE Autobackup at 16-OCT-03

piece handle=C:\ORACLE\ORA92\DATABASE\C-549349079-20031016-06 comment=NONE

Finished Control File and SPFILE Autobackup at 16-OCT-03

RMAN>

```
-- 4. Shutdown your database and remove the datafile associated with the backed
up
--   tablespace from its current directory.  I suggest a move rather than a
--   delete so that you have a backup recovery strategy should something go
wrong
--   with your RMAN recovery.
```

RMAN> shutdown

```
database closed
database dismounted
Oracle instance shut down
```

```
-- 5. Attempt to start the database.
```

RMAN> startup

```
connected to target database (not started)
Oracle instance started
database mounted
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of startup command at 10/16/2003 19:03:31
ORA-01157: cannot identify/lock data file 13 - see DBWR trace file
ORA-01110: data file 13: 'C:\ORACLE\ORADATA\DAVE\O1_MF_EXAMPLE_ZRTXR700_.DBF'
```

RMAN>

```
-- 6. Restore and recover the missing tablespace datafile.  Open the database.
```

RMAN> restore tablespace example;

Starting restore at 16-OCT-03

```
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=13 devtype=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: sid=14 devtype=DISK
channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00013 to C:\ORACLE\ORADATA\DAVE\O1_MF_EXAMPLE_ZRTXR700_.DBF
channel ORA_DISK_2: starting datafile backupset restore
channel ORA_DISK_2: specifying datafile(s) to restore from backup set
restoring datafile 00005 to C:\ORACLE\ORADATA\DAVE\EXAMPLE01.DBF
```

```
channel ORA_DISK_1: restored backup piece 1
piece handle=C:\ORACLE\BACKUPS\BACKUP_0PF425FS_1_1 tag=TAG20031016T185907
params
=NULL
channel ORA_DISK_1: restore complete
channel ORA_DISK_2: restored backup piece 1
piece handle=C:\ORACLE\BACKUPS\BACKUP_0QF425FS_1_1 tag=TAG20031016T185907
params
=NULL
channel ORA_DISK_2: restore complete
Finished restore at 16-OCT-03
```

```
RMAN> recover tablespace example;
```

```
Starting recover at 16-OCT-03
using channel ORA_DISK_1
using channel ORA_DISK_2
```

```
starting media recovery
media recovery complete
```

```
Finished recover at 16-OCT-03
```

```
RMAN> alter database open;
```

```
database opened
```

--Lesson 20. Logical Backups Workshops

```
-- Workshop: Logical Backup

-- 1. Produce a list of tables owned by the administrative user
--    created for this course.

  1 select table_name
  2 from dba_tables
  3* where owner = 'DAVE'
SQL> /

TABLE_NAME
-----
AUM_TEST_1
AUM_TEST_2
PLAN_TABLE
STORAGE_WORKSHOP1
T
T1

-- 2. Execute a consistent user-mode export on this user.

C:\>exp system/dave owner=dave consistent=y file=c:\oracle\backups\dave.dmp

Export: Release 9.2.0.1.0 - Production on Fri Oct 17 11:14:36 2003

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to: Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user DAVE
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user DAVE
About to export DAVE's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export DAVE's tables via Conventional Path ...
. . exporting table                AUM_TEST_1          26118 rows exported
```

```
. . exporting table          AUM_TEST_2          2259 rows exported
. . exporting table          PLAN_TABLE            0 rows exported
. . exporting table          STORAGE_WORKSHOP1      0 rows exported
. . exporting table          T                    29131 rows exported
. . exporting table          T1                     0 rows exported
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully without warnings.
```

C:\>

-- 3. Drop this user.

```
SQL> drop user dave cascade;
```

User dropped.

```
SQL>
```

-- 4. Take the necessary actions to recreate the user and all tables.

```
1 create user dave identified by dave
2 default tablespace users
3 quota unlimited on users
4* quota unlimited on tools
```

```
SQL> /
```

User created.

```
SQL> grant dba to dave;
```

Grant succeeded.

```
C:\>imp system/dave fromuser=dave file=c:\oracle\backups\dave.dmp
```

Import: Release 9.2.0.1.0 - Production on Fri Oct 17 11:18:58 2003

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to: Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

Export file created by EXPORT:V09.02.00 via conventional path
import done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
. importing SYSTEM's objects into SYSTEM
. importing DAVE's objects into DAVE
. . importing table "AUM_TEST_1" 26118 rows imported
. . importing table "AUM_TEST_2" 2259 rows imported
. . importing table "PLAN_TABLE" 0 rows imported
. . importing table "STORAGE_WORKSHOP1" 0 rows imported
. . importing table "T" 29131 rows imported
. . importing table "T1" 0 rows imported
Import terminated successfully without warnings.

--Lesson 21. SQL*Loader Workshop

```
-- Workshop:  SQL*Loader
```

```
-- 1. Run the supplied script SALES.SQL to create a table called SALES in your schema.
```

```
SQL> @sales
```

```
SQL> CREATE TABLE sales
```

```
 2      ( prod_id      NUMBER(6)
 3          CONSTRAINT sales_product_nn    NOT NULL
 4      , cust_id      NUMBER
 5          CONSTRAINT sales_customer_nn   NOT NULL
 6      , time_id      DATE
 7          CONSTRAINT sales_time_nn      NOT NULL
 8      , channel_id   CHAR(1)
 9          CONSTRAINT sales_channel_nn   NOT NULL
10      , promo_id     NUMBER(6)
11          CONSTRAINT sales_promo_nn    NOT NULL
12      , quantity_sold NUMBER(3)
13          CONSTRAINT sales_quantity_nn NOT NULL
14      , amount_sold  NUMBER(10,2)
15          CONSTRAINT sales_amount_nn   NOT NULL
16      )
17 /
```

Table created.

```
-- 2. Review the data found in the supplied file "SALES.DAT". Note that each field in the input file
```

```
-- is terminated by a "|" symbol.
```

```
-- 3. Use SQL*Loader to load all records in the SALES.DAT file into the SALES table.
```

```
LOAD DATA
APPEND
INTO TABLE sales
FIELDS TERMINATED BY "|"
( PROD_ID,
  CUST_ID,
  TIME_ID,
  CHANNEL_ID,
  PROMO_ID,
  QUANTITY_SOLD,
  AMOUNT_SOLD)
```

```
C:\OracleScripts\DBA>sqlldr dave/dave control=sales.ctl
```

-- 4. Query the SALES table to determine how many records were loaded. You should have 48,166 records.

```
SQL> select count(*) from sales;
```

```
      COUNT(*)  
-----
```

--Lesson 22. Oracle Networking Workshop

```
-- Workshop: Oracle Networking

-- 1. Backup your tnsnames.ora file

-- 2. Create a new connect descriptor in your local tnsnames.ora file. The
service name should be "CLASS" and it should connect you to the database you've
been using in class this week.

CLASS =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 10.0.0.9)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DAVE)
    )
  )
)

-- 3. Check the status of the listener.
lsnrctl
status
exit

-- 4. Stop the listener.
lsnrctl
stop

-- 5. Attempt connections using the tnsnames entry created in step 2. If you
are connecting to your database via
--       SQL*Net over a network, the connection will fail if the listener
is stopped. However, if your database is
--       on your local Windows machine the connection will probably still
work. This is because you are
--       connected to the database via a local Windows 'Bequeath'
connection, not via SQL*Net.
connect dave/dave@class

-- 6. Start the listener.
start

-- 7. Insure that you can connect to the database.
connect dave/dave@class

-- 8. Restore your original tnsnames.ora file.
```