Appendix B
Solutions

Solutions for Practice 2: Installing the Oracle Database Software

Background: In the practices of this course, you assume the role of a database administrator (DBA). The operating system (OS) accounts on your computer are:
- The `oracle` user with a password of `oracle`
- The `root` user with a password of `oracle`

The system administrator has set up the OS so that it is ready for the installation, and the installation media is staged at `/stage/Disk1`. Perform the following tasks as the default `oracle` OS user, unless otherwise indicated.

After installing the software, execute the `lab_02_05.sh` script, which creates a listener for you.

Note: Completing this practice is critical for all following practice sessions.

1. Install the Oracle database software as the `oracle` user. Navigate to the `/stage/Disk1` directory, and start the Oracle Universal Installer (OUI) by entering `./runInstaller`.

    a) Right-click your desktop and select **Open Terminal**, and then enter:

        $ cd /stage/Disk1
        $ ./runInstaller

2. Select your installation method for OUI.

    a) On the Installation Method page, select **Basic Installation**, and confirm the following settings:

Object	Setting
Database Home Location	`/u01/app/oracle/product/10.2.0/db_1`
Installation Type	Enterprise Edition
UNIX DBA Group	`oinstall`
Create Starter Database	*Deselected*

    Note: Ensure that you deselect the Create Starter Database option.

    b) Click Next.

    c) You are now on the page titled "Specify Inventory directory and credentials." Accept `/u01/app/oracle/oraInventory` as **inventory directory** and `oinstall` as **Operating System group name**. Click **Next**.

Solutions for Practice 2: Installing the Oracle Database Software (continued)

OUI is loading the products list and checking product-specific prerequisites.
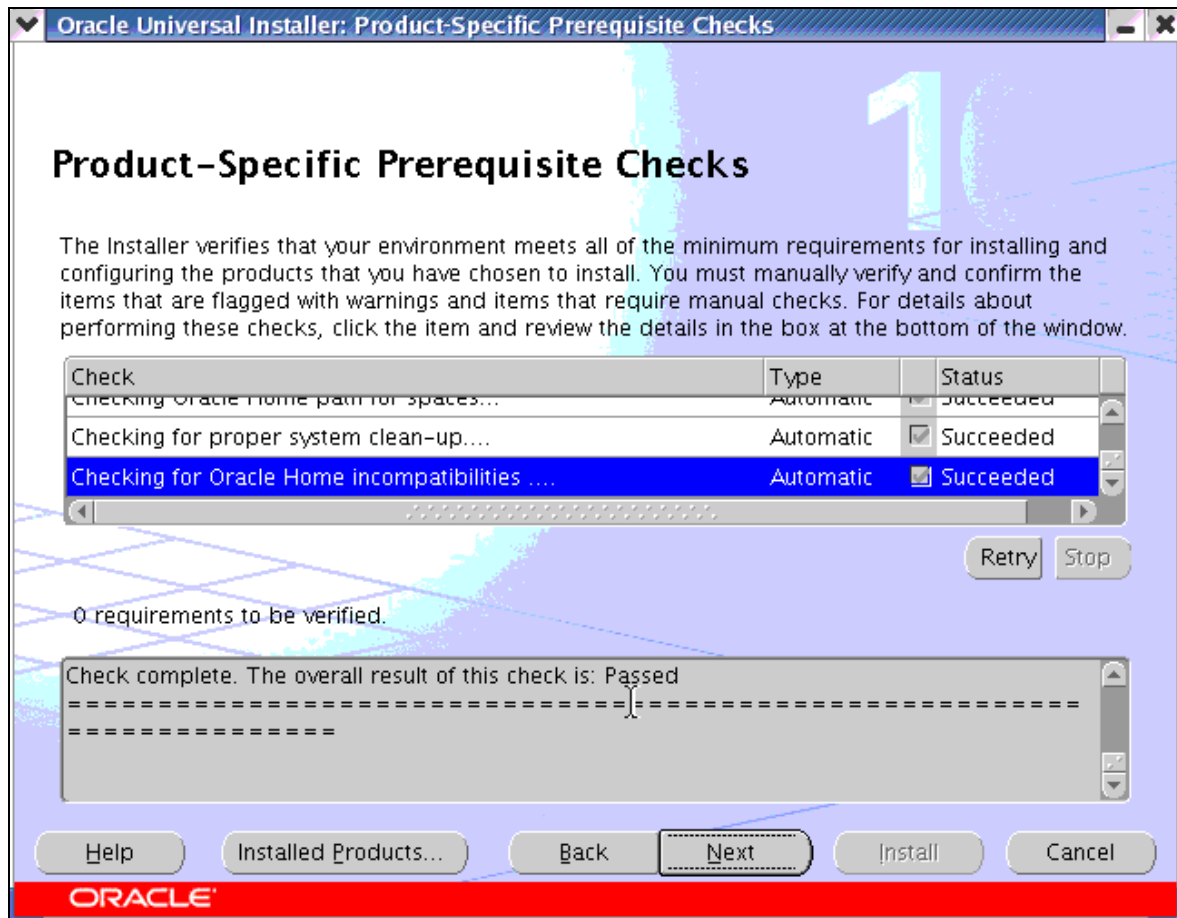
d) After OUI has finished its prerequisite checks on the Product-Specific Prerequisite Checks page, click **Next**.

3. When the prerequisite checks are finished, the Summary page is displayed.

a) Click **Install** to begin your installation.

Estimated installation time is 10–15 minutes. However, varying environments can greatly influence this estimate.

4. When the "Execute Configuration scripts" page appears, follow the instructions on that page, accept the default for the local `bin` directory, and then finish your installation with OUI.

Solutions for Practice 2: Installing the Oracle Database Software (continued)
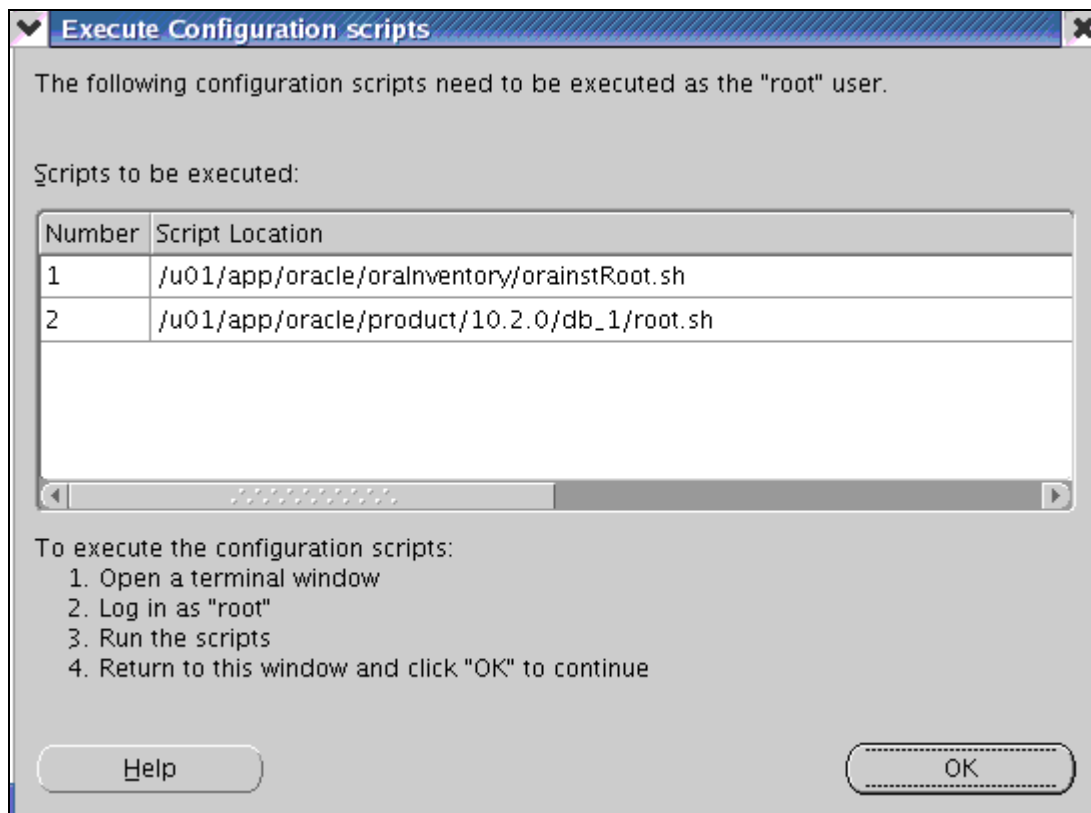
a) Right-click your desktop and select **Open Terminal**.

b) Run the `orainstRoot.sh` and `root.sh` scripts as `root`, and then click **Continue**.

```
$ su
# password: oracle <root password, does not appear on the screen>
# cd /u01/app/oracle/oraInventory
# ./orainstRoot.sh
# cd /u01/app/oracle/product/10.2.0/db_1
# ./root.sh
```

c) Accept the default for the local `bin` directory.

Solutions for Practice 2: Installing the Oracle Database Software (continued)

```
[oracle@EDRSR4P1 oracle]$ su
Password:
[root@EDRSR4P1 oracle]# cd /u01/app/oracle/oraInventory
[root@EDRSR4P1 oraInventory]# ./orainstRoot.sh
Changing permissions of /u01/app/oracle/oraInventory to 770.
Changing groupname of /u01/app/oracle/oraInventory to oinstall.
The execution of the script is complete
[root@EDRSR4P1 oraInventory]# cd /u01/app/oracle/product/10.2.0/db_1
[root@EDRSR4P1 db_1]# ./root.sh
Running Oracle10 root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/10.2.0/db_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
    Copying dbhome to /usr/local/bin ...
    Copying oraenv to /usr/local/bin ...
    Copying coraenv to /usr/local/bin ...


Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
[root@EDRSR4P1 db_1]# █
```
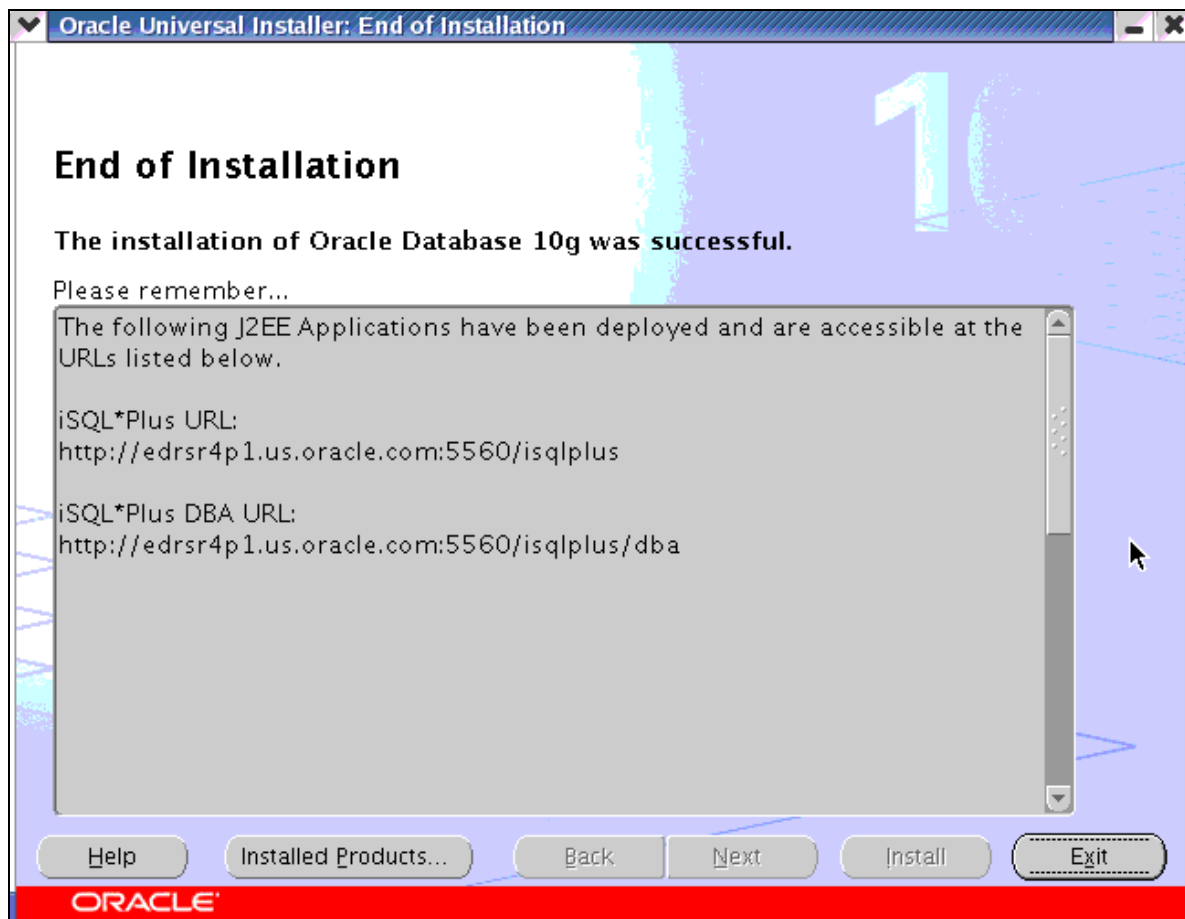
d) Enter **exit** to exit the `root` OS user.

e) Close the terminal window, and then click **OK** on the "Execute Configuration scripts" page.

The End of Installation page appears.

Solutions for Practice 2: Installing the Oracle Database Software (continued)

f) Make a note of your URLs. You will use them in later practice sessions.

*i*SQL*Plus URL:_____

*i*SQL*Plus DBA URL:_____

g) Optionally, click **Installed Products**, review the product inventory, then click **Close**.

h) Click **Exit**, and then click **Yes** to leave OUI.

5. Execute the `lab_02_05.sh` script, which is in the `/home/oracle/labs` directory. This script uses the `netca` utility to create a listener for you. If you create your first database with OUI (as part of your installation), then OUI invokes the network configuration assistant, which creates the first listener for you.

a) In a terminal window, enter:

```
cd /home/oracle/labs
./lab_02_05.sh
```

You see an activity log, which should end with a success message. If not, resolve any errors that might have occurred.

Solutions for Practice 3: Creating an Oracle Database

Background: You are about to begin creating your first Oracle database. You anticipate that several similar databases will be needed in the near future. Therefore, you decide to create your ORCL database, as well as a database template and the database creation scripts. Locate the scripts in the /home/oracle/labs directory (which is the directory that you use most often throughout this course).

After you create the ORCL database, you execute the lab_03_03.sh script, which creates a listener for you.

Note: Completing the database creation is critical for all following practice sessions.

1. Start the Database Configuration Assistant (DBCA).

    a) Open a terminal window as the oracle user; that is, right-click your desktop and select **Open Terminal**.

    b) To start the DBCA, enter:

```
$ dbca
```

2. Begin the ORCL database creation. Use the General Purpose database template.

    a) In the DBCA, click **Next** on the Welcome page.

    b) On the Operations page, select **Create a Database,** and then click **Next**.

    c) On the Database Templates page, select **General Purpose,** and then click **Show Details**.

    d) Review the template's details and answer the following questions.

    Question 1: How many control files are created?

    Answer: 3

    Question 2: Would it maximize database availability to multiplex them?

    Answer: Yes. (*This will be done in a later practice.*)

    Question 3: How many redo log groups are created?

    Answer: 3

    Question 4: Would it maximize database availability to mirror them?

    It depends: No, not in class, because there are already three groups and you have only one physical storage device; but yes, if you can put each group on a different physical storage device.

Solutions for Practice 3: Creating an Oracle Database (continued)

> *Question 5:* What is the database block size (db_block_size)?
>
> *Answer:* 8 KB
>
> *Question 6:* What is the value of Sample Schemas?
>
> *Answer:* Sample Schemas is set to False.
>
> **Note:** You will change this setting later in this practice to create the HR sample schema.
>
> *Question 7:* What is the template default for the Database Character Set?
>
> *Answer:* WE8ISO8859P1
>
> **Note:** You will change this setting later in this practice to use a Unicode database character set.
>
> e) Click **Close** to close the Template Details window.
>
> f) Click **Next**.

3. Create the ORCL database, as well as the ORCL template and the database generation scripts.

   a) On the Database Identification page, enter **orcl.oracle.com** as Global Database Name. The SID defaults to the database name **orcl**. Click **Next**.

   b) On the Management Options page, ensure that the following items are selected:

      - Configure the Database with Enterprise Manager
      - Use Database Control for Database Management

   c) Click **Next**.

   d) On the Database Credentials page, select **Use the Same Password for All Accounts** and enter **oracle** as Password and Confirm Password. Then, click **Next**.

   e) On the Storage Options page, select **File System**, and then click **Next**.

   f) On the Database File Locations page, select **Use Oracle-Managed Files**. Accept the default Database Area, and then click **Next**.

   g) On the Recovery Configuration page, select **Specify Flash Recovery Area,** and then click **Next**.

   h) On the Database Content page, select **Sample Schemas,** and then click **Next**.

   i) On the Memory tabbed page of the Initialization Parameters page, select **Custom** and then select **Automatic** for the **Shared Memory Management** setting.

**Solutions for Practice 3: Creating an Oracle Database (continued)**

j) On the Initialization Parameters Character Sets tabbed, select **Use Unicode (AL32UTF8)**.

k) Review the Sizing and Connection Mode tabbed pages, but do not change any values. Then, click **Next**.

l) On the Database Storage page, review your file names and locations. Then, click **Next**.

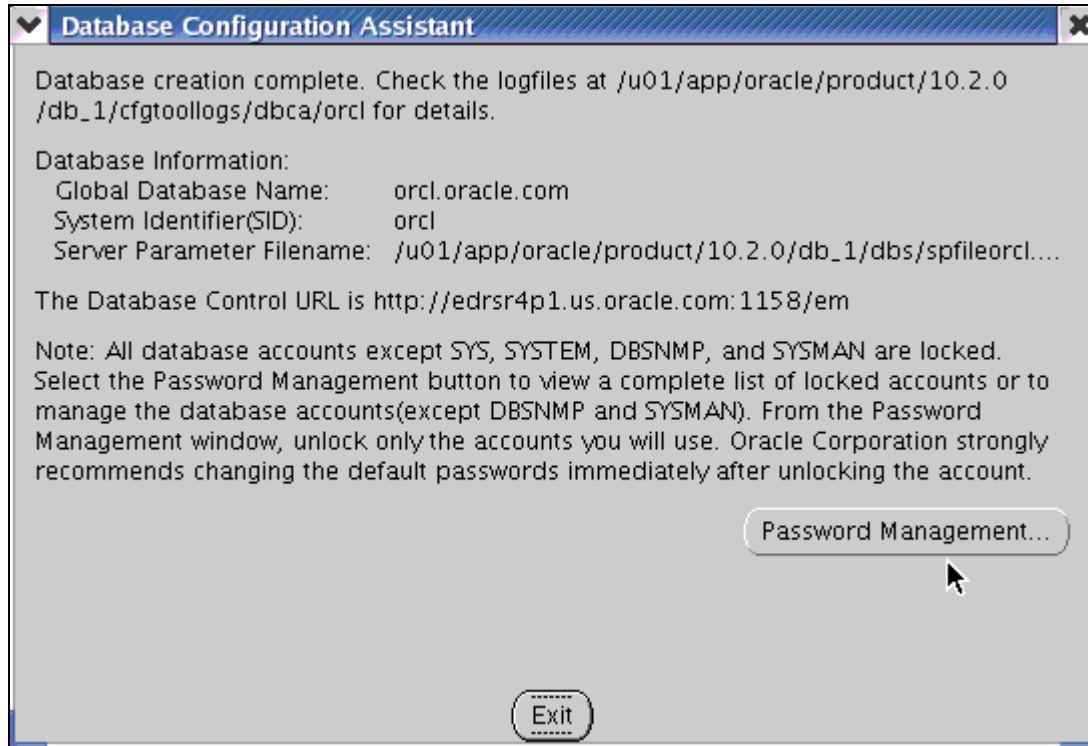**Solutions for Practice 3: Creating an Oracle Database (continued)**

m) On the Creation Options page, select **Create Database**.
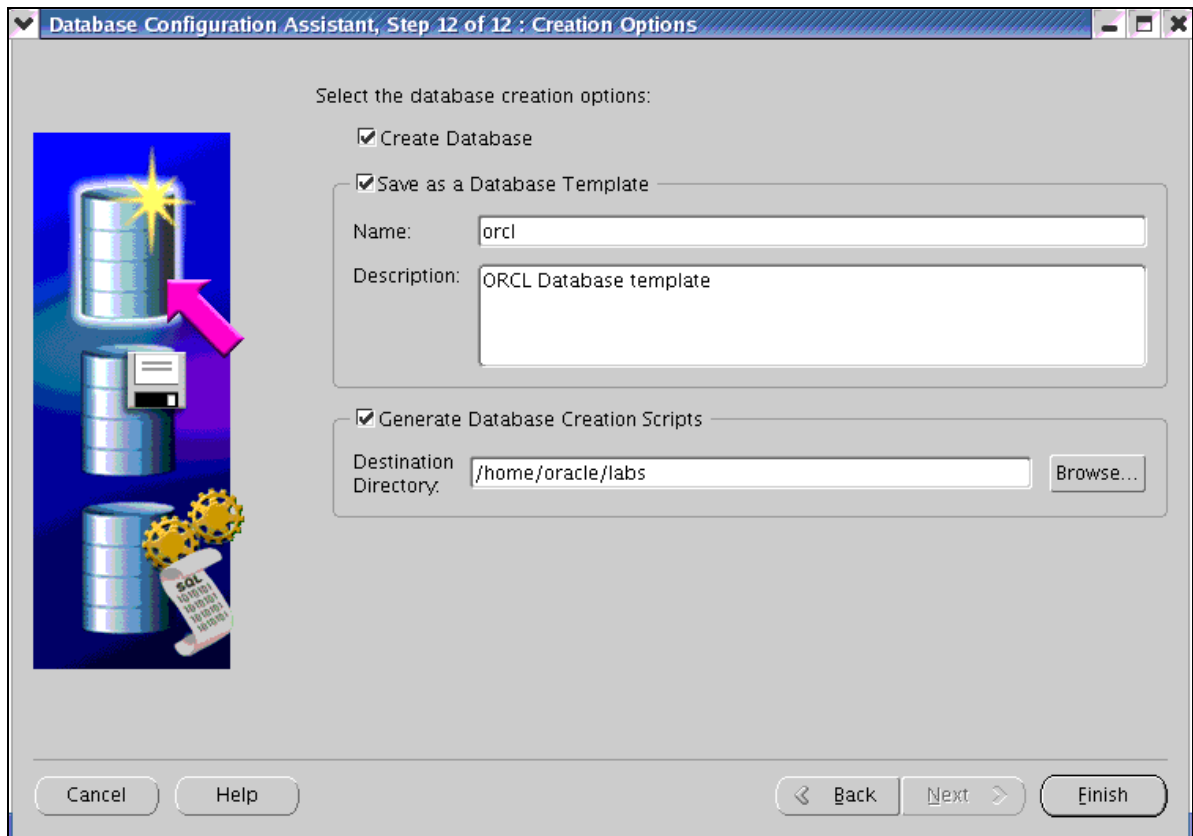
n) Optionally, select all creation options and enter `orcl` as **Name** for the database template, `ORCL Database template` as **Description**, and `/home/oracle/labs` as **Destination Directory**. Then, click Finish.

o) The Confirmation page appears. Review options and parameters, such as `Sample Schemas (true)`, `db_block_size (8KB)`, `sga_target (270MB)`, `undo_management (AUTO)`, `Database Character Set (AL32UTF8)`, and then click **OK**.

p) Click **OK** to acknowledge that the template has been created. Then, acknowledge the generation of the database scripts (if you selected those options).

Solutions for Practice 3: Creating an Oracle Database (continued)

q) The DBCA displays the progress of the various installation steps. When the database itself has been created, the DBCA displays essential information about the database. Make note of this information. The Database Control URL will be used in several of the following practice sessions.

r) Click the **Password Management** button.

s) Scroll down the Password Management page until you see the HR User Name.

Solutions for Practice 3: Creating an Oracle Database (continued)

t) Deselect **Lock Account?** and enter `hr` as New Password and Confirm Password. Then, click **OK**.

User Name	Lock Account?	New Password	Confirm Password
SI_INFORMTN_SCHEMA	✔		
OLAPSYS	✔		
MDDATA	✔		
IX	✔		
SH	✔		
DIP	✔		
OE	✔		
SCOTT	✔		
HR		**	**
TSMSYS	✔		
BI	✔		
PM	✔		

u) Click **Exit** to close the DBCA.

You completed your task to create a database and (optionally) a database template and database generation scripts.

Solutions for Practice 4: Managing the Oracle Instance

Background: You have just installed the Oracle software and created a database. You want to ensure that you can start and stop the database and see the application data.

1. Invoke Enterprise Manager, and log in as the SYS user. Which port number does this database use? You noted this in Practice 3.

    Answer: 1158

    a) Open a Web browser.

    b) If using Mozilla, the Select User Profile dialog box appears.
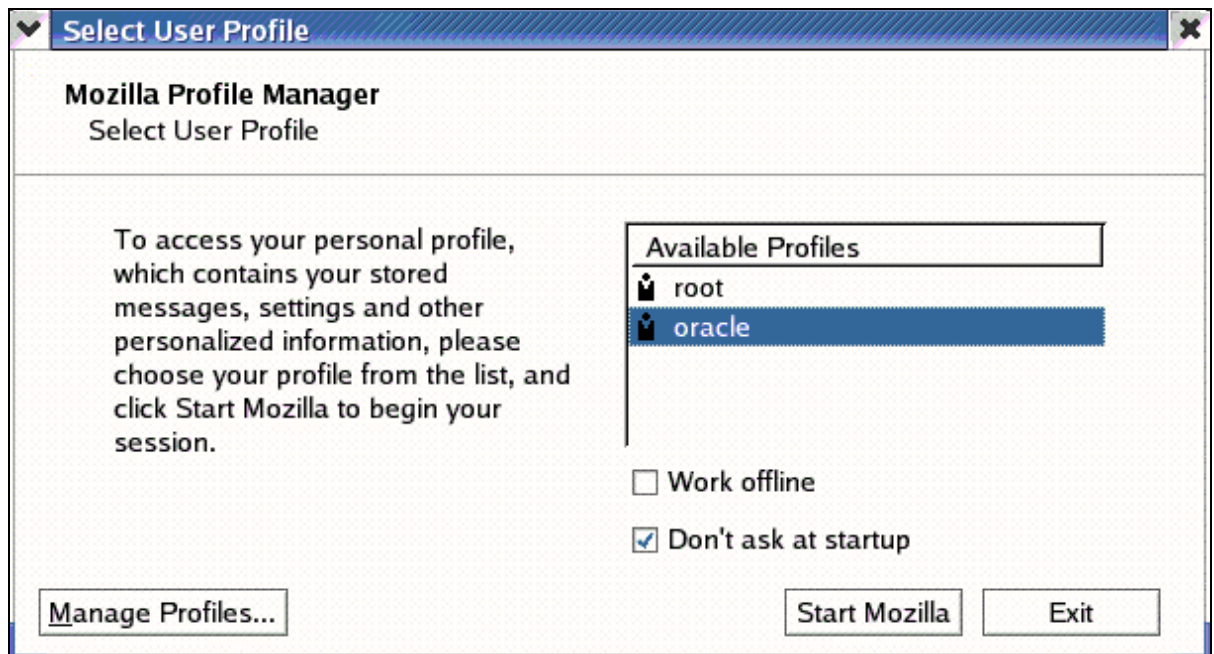
    Select **oracle** as the profile, and select **Don't ask at startup**, and then click **Start Mozilla**.
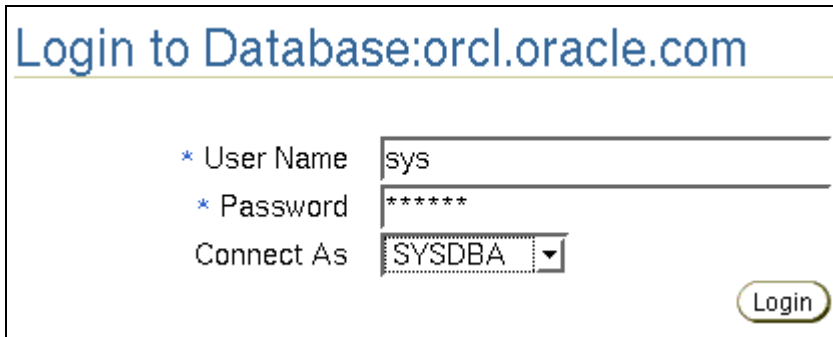
    c) Enter the URL that you wrote down in Practice 3. It has the following format:

    http://*hostname*:*portnumber*/em

    The Oracle Enterprise Manager window appears.

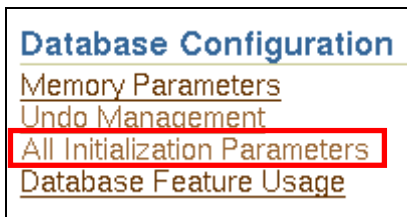**Solutions for Practice 4: Managing the Oracle Instance (continued)**

    d)   Log in by entering `sys` as **User Name** and `oracle` as **Password**, and selecting `SYSDBA` as **Connect As**. Then, click **Login**.
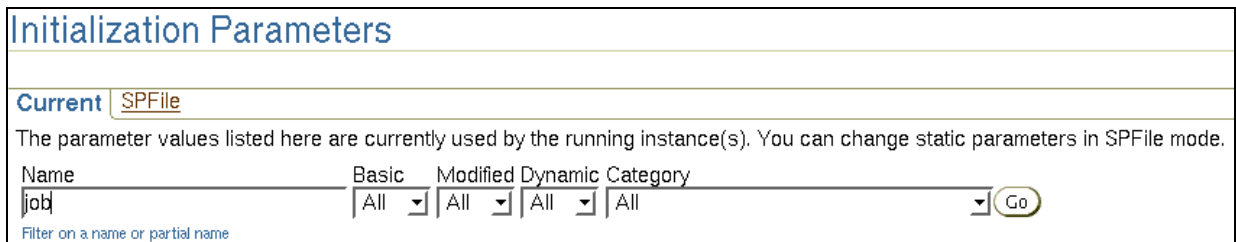

    e)   You may receive a security warning. In class, deselect **Alert me whenever I submit information that's not encrypted.**, and then click **Continue**.

        The first time you log in after installing the software, the Oracle Database 10*g* Licensing Information page appears.

    f)   To acknowledge this information, click **I agree** at the bottom of the page.

2. View the initialization parameters and set the `JOB_QUEUE_PROCESSES` parameter to `15`. What SQL statement is run to do this?

    a)   Select **Administration** > Database Administration > Database Configuration > **All Initialization Parameters**.
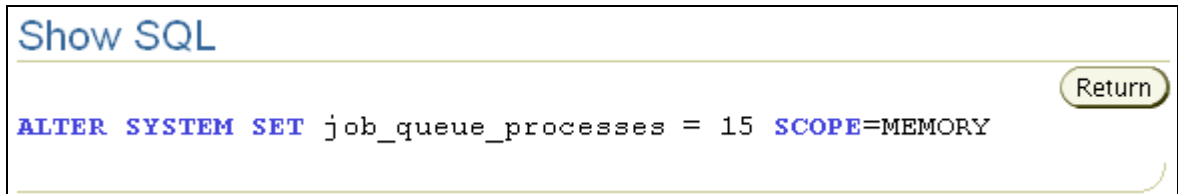

    b)   Enter `job` in the **Name** field, and then click **Go**.

    c)   When the `JOB_QUEUE_PROCESSES` initialization parameter appears, change its value from 10 to 15.

Solutions for Practice 4: Managing the Oracle Instance (continued)

     d) Click **Show SQL** and note the SQL statement that is going to be run.

> Show SQL
>
>   (Return)
>
> `ALTER SYSTEM SET job_queue_processes = 15 SCOPE=MEMORY`

     e) Click **Return**, and then click **Apply**.

3. *Question:* What is the significance of a check in the Dynamic column?

   Answer: A "dynamic" parameter can be modified while the database is active.

4. Shut down the database instance by using Enterprise Manager.

   Question: What SQL is executed to do this?

     a) In the Enterprise Manager browser session, click the **Database** tab.

     b) Click **Shutdown**.

     c) For **Host Credentials**, enter `oracle` as **Username** and `oracle` as **Password**.

     d) Click OK.

        The Startup/Shutdown: Confirmation page appears.

     e) Click **Advanced Options** to see the mode for shutting down, but do not change the mode; it should remain as "Immediate."

     f) Click **Cancel** to return to the previous page.

     g) Click **Show SQL** to view the SQL that is going to be executed to perform the shutdown operation.

   Question: What SQL is executed to do this?

   Answer: SHUTDOWN IMMEDIATE

> Show SQL
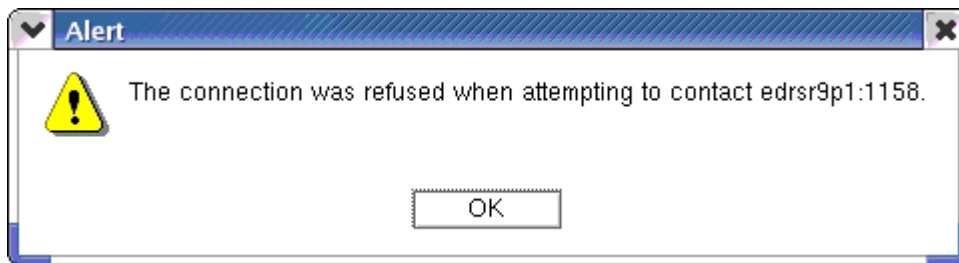>
>   (Return)
>
> `SHUTDOWN immediate`

     h) Click **Return**.

Solutions for Practice 4: Managing the Oracle Instance (continued)

i) Click **Yes** to confirm the shutdown operation.

j) Click Refresh. If you see the following error during the refresh, click **OK** and continue to refresh. The error will resolve itself.

k) Note that the **Status** of the instance is now "Down."

5. Using SQL*Plus, verify that you are <u>not</u> able to connect as the HR user to a database that has been shut down.

a) In the Linux command window, enter the following to attempt to log in to the database:

```
$ sqlplus hr
```

b) Enter hr for the password.

c) Note the "ORACLE not available" error message.

d) Press [Ctrl], [D] to exit the username prompt.

6. Use Enterprise Manager to restart the database instance, and then log in as the SYS user again.

Question: What SQL is run to accomplish the database startup?

a) In Enterprise Manager, click the **Startup** button.

b) Enter oracle for both **Username** and **Password** in the **Host Credentials** region.

c) Click **OK**.

Solutions for Practice 4: Managing the Oracle Instance (continued)

    d)   The Startup/Shutdown: Confirmation page appears. Click **Show SQL** to view the SQL that is about to run.

    Question: What SQL is run to accomplish the database startup?

    Answer: STARTUP

Show SQL
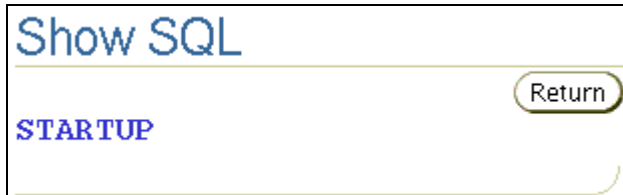
Return

STARTUP

    e)   Click **Return**.

    f)   Click **Yes** to confirm the startup operation.

    g)   The Startup/Shutdown: Activity Information page appears. Wait for the login page to appear, at which time you can log in as SYS user with the oracle password and the SYSDBA privilege.

7. In the alert log, view the phases that the database went through during startup. What are they?

    a)   Select **Database** > Related Links > **Alert Log Content**.

Solutions for Practice 4: Managing the Oracle Instance (continued)

b) Scroll toward the bottom of the log and review the phases of the database during startup. Your alert log may look different from this screenshot, based on different system activities.


```
Fri May 20 09:47:26 2005
Database mounted in Exclusive Mode
Completed: ALTER DATABASE   MOUNT
Fri May 20 09:47:26 2005
ALTER DATABASE OPEN
Fri May 20 09:47:26 2005
Thread 1 opened at log sequence 209
  Current log# 1 seq# 209 mem# 0: /u01/app/oracle/oradata/orcl/redo01.log
Successful open of redo thread 1
Fri May 20 09:47:26 2005
MTTR advisory is disabled because FAST_START_MTTR_TARGET is not set
Fri May 20 09:47:26 2005
SMON: enabling cache recovery
Fri May 20 09:47:27 2005
Successfully onlined Undo Tablespace 1.
Fri May 20 09:47:27 2005
SMON: enabling tx recovery
Fri May 20 09:47:27 2005
Database Characterset is AL32UTF8
replication_dependency_tracking turned off (no async multimaster replication found)
Starting background process QMNC
QMNC started with Fri May 20 09:47:30 2005
db_recovery_file_dest_size of 2048 MB is 0.00% used. This is a
user-specified limit on the amount of space that will be used by this
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
Fri May 20 09:47:33 2005
Completed: ALTER DATABASE OPEN
```

c) Note that the modes the database goes through during startup are MOUNT and OPEN. Click **OK** to close the alert log.

8. Test access to *i*SQL*Plus for your HR application developers. (Navigation aid: Database > *i*SQL*Plus). Use the Normal role, hr username and password, and the default setting as Connect Identifier. If there is an error accessing *i*SQL*Plus, then start the isqlplus process using the isqlplusctl start command at the OS prompt, and then reattempt. After connecting, select the contents of the EMPLOYEES table.

a) Select **Database** > Related Links > **iSQL*Plus**.

The *i*SQL*Plus Connection Role page appears.

b) Notice that the SYSOPER and SYSDBA roles require special setup and authentication for security reasons. Select **Normal**, and then click **Continue**.

Solutions for Practice 4: Managing the Oracle Instance (continued)

c) If you see an error message saying that the connection was refused, that means you need to start the supporting process. To do this, enter the `isqlplusctl start` command at the OS prompt, and then retry step (b).

```
$ isqlplusctl start
iSQL*Plus 10.2.0.1.0
Copyright (c) 2003, 2005, Oracle.  All rights reserved.
Starting iSQL*Plus ...
iSQL*Plus started.
```

d) On the Login page, enter hr as **Username** and **Password** and leave the **Connect Identifier** set to its default value. Click the **Login** button.

The Confirm window appears, with the Password Manager offering to remember this login for you. Click **No**.

e) Enter the following statement in the workspace, and then click the **Execute** button:

```
SELECT * FROM EMPLOYEES;
```

The content of the EMPLOYEES table appears in a tabular form.

f) When you have finished reviewing the information, click **Logout** in the top-right corner of the page.

g) Click X in the top-right window frame to close the *i*SQL*Plus window.

Solutions for Practice 5: Managing Database Storage Structures

Background: You need to create a new tablespace for the INVENTORY application. You also need to create a database user that is not as privileged as the SYS user.

1. Enter ./lab_05_01.sh to run a script that creates the DBA1 user. It is located at /home/oracle/labs. The password for DBA1 is oracle.

   a) Start a Linux command shell by right-clicking your desktop and selecting **Open Terminal**.

   b) Change the current directory to the labs directory by entering:

   ```
   $ cd labs
   ```

   c) Enter the following command to run the script that creates the DBA1 user:

   ```
   $ ./lab_05_01.sh
   ```

   d) Leave the command shell window open. You will use it again later.

2. Use the Setup link in the top-right corner of Enterprise Manager (EM) to define the DBA1 user as one who can perform administrative tasks in EM. When the non-SYS user is configured, log out as the SYS user and log in as the DBA1 user. Use the DBA1 user to perform the rest of these tasks, unless otherwise indicated.

   a) In the far top-right corner of the EM window, click **Setup**.

   b) Click **Create** to add the DBA1 user to the Administrators list. This will enable the DBA1 user to perform management tasks by using Enterprise Manager.
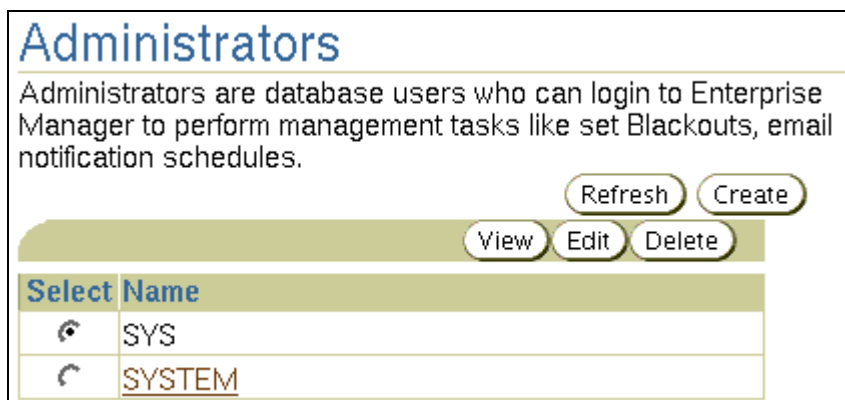
Solutions for Practice 5: Managing Database Storage Structures (continued)

   c) Enter `dba1` as **Name**, and `oracle` as **Password** and **Confirm Password**. Leave **Email Address** blank, and leave **Super Administrator** selected, and then click **Finish**.

   d) On the Create Administrator: Review page, click **Finish** again.

   e) Now that there is a non-`SYS` user, click **Logout** in the top-right corner, and then click **Login**.

   f) Enter `DBA1` as **User Name** and `ORACLE` as **Password**, and select `SYSDBA` as **Connect As**. Then, click **Login**.

     The first time you log in as a new user, the licensing page appears.

   g) To acknowledge this information, click **I agree** at the bottom of the page.

     The Database Home page appears.

3. Using Enterprise Manager, view information about the `EXAMPLE` tablespace. Answer the following questions about it:

   a) In Enterprise Manager, select **Administration** > Database Administration **> Storage >** **Tablespaces**.

   b) Click the `EXAMPLE` tablespace name.

   c) *Question 1:* What percentage of free space can be used up before the Warning threshold is reached?

Answer: 85%

Solutions for Practice 5: Managing Database Storage Structures (continued)

d) From the **Actions** drop-down list, select **Show Tablespace Contents**, and then click **Go**.

e) The Show Tablespace Contents page appears.

Segment Name	Type	Size (KB) ▽	Extents
SH.CUSTOMERS	TABLE	12,288	27
SH.SUPPLEMENTARY_DEMOGRAPHICS	TABLE	4,096	19
OE PRODUCT DESCRIPTIONS	TABLE	3,072	18

⊖ Previous 1-10 of 418 ▾ Next 10 ⊖

f) *Question 2:* How many segments are there in the EXAMPLE tablespace?

Answer: 418

g) Select INDEX from the **Type** drop-down list in the Search region, and then click **Go**.

Segment Name	Type	Size (KB) ▽	Extents
SH.CUSTOMERS_PK	INDEX	1,024	16
OE.PROD_NAME_IX	INDEX	512	8
OE PRD DESC PK	INDEX	320	5

⊖ Previous 1-10 of 77 ▾ Next 10 ⊖

h) *Question 3:* Which index in the EXAMPLE tablespace takes up the most space?

Answer: SH.CUSTOMERS_PK

Question 4: Which segment is stored physically first in the tablespace? That is, which one is stored right after the tablespace header?

i) Scroll to the bottom of the page, and then click the plus icon to the left of the Extent Map label.

j) After several seconds, the extent map appears. Note that the map legend indicates that pink is the tablespace header.

k) Scroll back to the top of the page, select **All Types** from the **Type** drop-down list, and then click **Go**.

Solutions for Practice 5: Managing Database Storage Structures (continued)

l) Click the extent just to the right of the tablespace header extent.
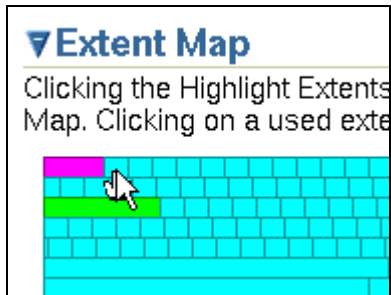
m) Scroll to the top of the page again, and note the segment that is being pointed to:

○	PM.SYS_LOB0000051820C00004$$	LOBSEGMENT	128	2
○	PM.SYS_LOB0000051819C00015$$	LOBSEGMENT	128	2
➡	HR.REGIONS	TABLE	64	1
○	HR.LOCATIONS	TABLE	64	1

Answer: HR.REGIONS

4. Create a new, locally managed tablespace (LMT) called INVENTORY of size 5 MB.

   Note: The INVENTORY tablespace will be used in later practice sessions during this class.

   a) In Enterprise Manager, select **Administration** > Database Administration > Storage > **Tablespaces**.

   b) Click **Create**.

   c) Enter INVENTORY as the tablespace name, and verify that **Extent Management** is **Locally Managed**, **Type** is **Permanent**, **Status** is **Read Write**, and **Use Bigfile tablespace** is <u>not</u> selected.

   d) Click **Add** in the Datafiles region.

   e) On the Add Datafile page, enter inventory01.dbf for **File Name**, and 5 MB as **File Size**.

   f) Click **Continue**.

   g) Click the **Storage** tab, and verify that **Extent Allocation** is **Automatic**, **Segment Space Management** is **Automatic**, and **Logging** is enabled.

   h) Click the **General** tab.

   i) Click **Show SQL** to see the SQL that will be run, and then click **Return**.

   j) Click **OK**, and a successful Update Message appears.

Solutions for Practice 5: Managing Database Storage Structures (continued)

5. As the DBA1 user, run the `lab_05_05.sql` script to create and populate a table (called X) in the INVENTORY tablespace. What error do you eventually see?

   a) Right-click the desktop and select **Open Terminal** to start a command shell. Then enter:

   ```
   $ cd labs
   ```

   b) Enter the following to run the script:

   ```
   $ sqlplus dba1/oracle @lab_05_05.sql
   ```

   c) Note that there is eventually an error ORA-01653 stating that the table cannot be extended. There is not enough space to accommodate all of the rows to be inserted.

   ```
   1024 rows created.

   SQL> insert into x select * from x
     2  /
   insert into x select * from x
   *
   ERROR at line 1:
   ORA-01653: unable to extend table DBA1.X by 128 in tablespace INVENTORY


   SQL> commit
     2  /

   Commit complete.
   ```

6. Go to the Enterprise Manager window and define space for 50 MB in the tablespace instead of 5 MB, while keeping the same single data file in the tablespace. What is the ALTER statement that is executed to make this change?

   a) Select **Administration** > Database Administration **>** Storage > **Tablespaces**.

   b) Select the INVENTORY tablespace, and then click **Edit**.

   c) In the **Datafiles** region, click **Edit**.

   d) Change **File Size** from 5 MB to 50 MB.

   e) Click **Continue** to return to the General tabbed page.

   f) Click **Show SQL** to see the SQL that will be run. Note that it is an ALTER DATABASE statement. Click **Return**.

Solutions for Practice 5: Managing Database Storage Structures (continued)

```
Show SQL
                                                          (Return)
ALTER DATABASE DATAFILE
'/u01/app/oracle/oradata/ORCL/datafile/inventory01.dbf'
RESIZE 50M
```

     g) Click **Apply**.

7. Go back to the terminal window and run the `lab_05_07.sql` script. It drops the table and reexecutes the original script that previously returned the space error.

     a) Go to the SQL*Plus window.

     b) Enter the following to run the script:

```
$ sqlplus dba1/oracle @lab_05_07.sql
```

     c) Note that the same number of row inserts are attempted, and there is no error because of the increased size of the tablespace.

8. In an terminal window, run the `lab_05_08.sql` script to clean up the tablespace for later practice sessions.

     a) Enter the following to run the script:

```
$ sqlplus dba1/oracle @lab_05_08.sql
```

Solutions for Practice 6: Administering User Security

Background: You need to create a user account for Jenny Goodman, the new human resources department manager. There are also two new clerks in the human resources department, David Hamby and Rachel Pandya. All three of them must be able to log in to the ORCL database, select data from, and update records in the HR.EMPLOYEES table. The manager also needs to be able to insert and delete new employee records. Ensure that if the new users forget to log out at the end of the day, they will automatically be logged out after 15 minutes. You also need to create a new user account for the inventory application that you are installing.

1. **Mandatory task:** Review and run the lab_06_01.sh script (located in the /home/oracle/labs directory) to create the INVENTORY user, which you will use in the next practice.

   a) In a terminal window, enter:

   ```
   cd /home/oracle/labs
   more lab_06_01.sh
   ./lab_06_01.sh
   ```

2. Create a profile named HRPROFILE, allowing 15 minutes idle time.

   a) Invoke Enterprise Manager as the DBA1 user in the SYSDBA role for your ORCL database.

   b) Select **Administration** > Schema > Users & Privileges > **Profiles**.

   c) Click the **Create** button.

   d) Enter **HRPROFILE** in the **Name** field.

   e) Enter **15** in the **Idle Time (Minutes)** field.

Solutions for Practice 6: Administering User Security (continued)

f) Leave all the other fields set to DEFAULT.

g) Click the **Password** tab, and review the Password options, which are currently all set to DEFAULT.

h) Optionally, click the **Show SQL** button, review your underlying SQL statement, and then click **Return**.

i) Finally, click **OK** to create your profile.

3. Set the RESOURCE_LIMIT initialization parameter to TRUE so that your profile limits will be enforced.

a) Select **Administration** > Database Administration > Database Configuration > **All Initialization Parameters**.

b) Enter **RESOURCE_LIMIT** in the **Filter** field, and then click **Go**.

c) Select **TRUE** from the **Value** drop-down list, and then click **Apply**.

Solutions for Practice 6: Administering User Security (continued)

Initialization Parameters

Show SQL   Revert   Apply

Current | SPFile

The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFile mode.

Name
RESOURCE_LIMIT

Basic [All ▾] Modified [All ▾] Dynamic [All ▾] Category [All ▾] Go

Filter on a name or partial name

☐ Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Save to File

Name △	Help	Revisions	Value	Comments	Type	Basic	Modified	Dynamic	Category
resource_limit	ⓘ		TRUE ▾		Boolean			✔	Resource Manager

4. Create the role named HRCLERK with SELECT and UPDATE permissions on the HR.EMPLOYEES table.

   a) Select **Administration** > Schema > Users & Privileges > **Roles**.

   b) Click the **Create** button at the top-right of the page.

   c) Enter **HRCLERK** in the **Name** field. This role is not authenticated.

   d) Click **Object Privileges**.

   e) Select **Table** from the **Select Object Type** drop-down list, and then click **Add**.

   f) Enter **HR.EMPLOYEES** in the **Select Table Objects** field.

   g) Move the SELECT and UPDATE privileges to the **Selected Privileges** box. Click **OK**.

   h) Click the **Show SQL** button, and review your underlying SQL statement.

```
CREATE ROLE "HRCLERK" NOT IDENTIFIED
GRANT SELECT ON "HR"."EMPLOYEES" TO "HRCLERK"
GRANT UPDATE ON "HR"."EMPLOYEES" TO "HRCLERK"
```

   i) Click **Return**, and then click **OK** to create the role.

Solutions for Practice 6: Administering User Security (continued)

5. Create the role named `HRMANAGER` with `INSERT` and `DELETE` permissions on the `HR.EMPLOYEES` table. Grant the `HRCLERK` role to the `HRMANAGER` role.

   a) Select **Administration** > Schema > Users & Privileges > **Roles**.

   b) Click **Create**.

   c) Enter **HRMANAGER** in the **Name** field. This role is not authenticated.

   d) Click **Object Privileges**.

   e) Select **Table** from the **Select Object Type** drop-down list, and then click **Add**.

   f) Enter **HR.EMPLOYEES** in the **Select Table Objects** field.

   g) Move the `INSERT` and `DELETE` privileges to the **Selected Privileges** box. Click **OK**.

   h) Click **Roles**, and then click **Edit List**.

   i) Move the `HRCLERK` role into the **Selected Roles** box, and then click **OK**.

   j) Click the **Show SQL** button, and review your underlying SQL statement.

```
CREATE ROLE "HRMANAGER" NOT IDENTIFIED
GRANT DELETE ON "HR"."EMPLOYEES" TO "HRMANAGER"
GRANT INSERT ON "HR"."EMPLOYEES" TO "HRMANAGER"
GRANT "HRCLERK" TO "HRMANAGER"
```

   k) Click **Return**, and then click **OK** to create the role.

Solutions for Practice 6: Administering User Security (continued)

6. Create an account for David Hamby, a new HR clerk.

   a) Select **Administration** > Schema > Users & Privileges > **Users**.

   b) Click **Create**, and enter **DHAMBY** in the Name field.

   c) Select **HRPROFILE** for the Profile.

   d) Select Password Authentication, and enter **newuser** as password. Enter it into the **Confirm Password** field also. Select the **Expire Password now** check box so that David will have to change the password the first time he logs in.

   e) Click **Roles**. Notice that the CONNECT role has automatically been assigned to the user.

   f) Add the HRCLERK role by clicking **Edit List** and moving the HRCLERK role into the **Selected Roles** box. Click **OK** to close the Modify Roles window.

   g) Click **OK** again to create the user.

7. Create an account for Rachel Pandya, another new HR clerk. Repeat the steps under step 6 with RPANDYA as the username.

Solutions for Practice 6: Administering User Security (continued)

8. Create an account for Jenny Goodman, the new HR manager. Repeat the steps under step 6 with `JGOODMAN` as the username and selecting the `HRMANAGER` role instead of the `HRCLERK` role.

   a) Click the **Show SQL** button and review your underlying SQL statement.

   ```
   CREATE USER "JGOODMAN" PROFILE "HRPROFILE" IDENTIFIED BY "*******" PASSWORD
   EXPIRE ACCOUNT UNLOCK
   GRANT "CONNECT" TO "JGOODMAN"
   GRANT "HRMANAGER" TO "JGOODMAN"
   ```

   b) Click **Return**, and then click **OK** to create the user.

9. Test the new users in SQL*Plus. Connect to the `ORCL` database as the `DHAMBY` user. Use `oracle` as the new password. Select the row with `EMPLOYEE_ID=197` from the `HR.EMPLOYEES` table. Then, attempt to delete it. (You should get the "insufficient privileges" error.)

   a) In a terminal window, enter:

   ```
   sqlplus dhamby/newuser
   ```

   or if you are already in SQL*Plus, use the `CONNECT` command. If you reconnect as `dhamby` in SQL*Plus, the login and change-of-password session looks like this:

   ```
   SQL> CONNECT dhamby/newuser

   ERROR:
   ORA-28001: the password has expired

   Changing password for dhamby
   New password: oracle      <<<Password does not appear on screen
   Retype new password: oracle    <<<Password does not appearon screen
   Password changed

   Connected to:
   Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 – Production
   With the Partitioning, OLAP and Data Mining options

   SQL>
   ```

   b) Select the salary for `EMPLOYEE_ID=197` from the `HR.EMPLOYEES` table.

   ```
   SQL> SELECT salary FROM hr.employees WHERE EMPLOYEE_ID=197;

          SALARY
       ----------
            3000
   ```

Solutions for Practice 6: Administering User Security (continued)

c) Now attempt to delete the same record from the `hr.employees` table.

```
SQL> DELETE FROM hr.employees WHERE EMPLOYEE_ID=197;
DELETE FROM  hr.employees WHERE EMPLOYEE_ID=197
                *
ERROR at line 1:
ORA-01031: insufficient privileges
```

10. Repeat the delete attempt as the `JGOODMAN` user. After deleting the row, issue a rollback, so that you still have the original 107 rows.

a) Connect to the `ORCL` database as the `JGOODMAN` user.

```
SQL> connect jgoodman/newuser
ERROR:
ORA-28001: the password has expired
<Change the password as shown above>
```

b) Select the row with `EMPLOYEE_ID=197` from the `HR.EMPLOYEES` table.

```
SQL> SELECT salary FROM hr.employees WHERE EMPLOYEE_ID=197;

    SALARY
  ----------
      3000
```

c) Now delete the same row from the `HR.EMPLOYEES` table.

```
SQL> DELETE FROM  hr.employees WHERE EMPLOYEE_ID=197;

1 row deleted.
```

d) Roll back the delete operation (because this was just a test).

```
SQL> rollback;

Rollback complete.
```

e) Confirm that you still have 107 rows in this table.

```
SQL> SELECT COUNT(*) FROM hr.employees;

  COUNT(*)
  ----------
       107

SQL>
```

Question 1: Where was the row stored after deletion?

Answer: It was stored in the Undo tablespace.

Solutions for Practice 6: Administering User Security (continued)

Question 2: When you created the new users, you did not select a default or temporary tablespace. What determines the tablespaces that the new users will use?

Answer: The system-defined default permanent and temporary tablespaces

Question 3: You did not grant the CREATE SESSION system privilege to any of the new users, but they can all connect to the database. Why?

Answer: Because Enterprise Manager automatically assigns the CONNECT role to the new users, and CREATE SESSION is contained within that role

11. Review the lab_05_01.sql script and the lab_05_01.txt log file that it generated when you created the DBA1 user.

    a) Double-click the **oracle's Home** icon on your desktop.

    b) Navigate to the **labs** directory.

    c) Double-click the **lab_05_01.sql** file, and review its content.

    d) When you have finished reviewing the file, click the **Up** icon to return to the labs directory.

    e) Now, double-click the **lab_05_01.txt** file, which was created when you executed the lab_05_01.sql file.

    f) When you have finished reviewing the file, click the X (Close Window) icon.

12. Use SQL*Plus to connect to the ORCL database as the RPANDYA user. Change the password to oracle. (You must change the password, because this is the first connection as RPANDYA.) Leave RPANDYA connected during the next lesson or at the end of the day. HRPROFILE specifies that users whose sessions are inactive for more than 15 minutes will automatically be logged out. Verify that the user was automatically logged out by trying to select from the HR.EMPLOYEES table again.

```
ERROR at line 1:
ORA-02396: exceeded maximum idle time, please connect again
```

Solutions for Practice 7: Managing Schema Objects

Background: You need to create schema objects for the new inventory application. Work as the DBA1 user in the SYSDBA role for your ORCL database.

1. Return to the Enterprise Manager browser session, or invoke EM as the DBA1 user in the SYSDBA role for your ORCL database.

2. In the INVENTORY tablespace, create the PRODUCT_MASTER table in the INVENTORY schema. The specifications of the table are:

   > PRODUCT_ID number(7). This is the primary key field. (Constraint name: PK_INV)
   > PRODUCT_NAME varchar2(50) with a Not NULL constraint
   > CODE varchar2(10) with a Not NULL constraint
   > REORDER_THRESHOLD number(5) with a check constraint ensuring that the number is always greater than zero (Constraint name: CHK_REORDER)
   > COST number(5,2)
   > PRICE number(5,2)

   a) Select **Administration** > Schema > Database Objects > **Tables**.

   b) Click **Create**.

   c) Leave the default table type to be **Standard, Heap Organized permanent**, and then click **Continue**.

   d) Enter PRODUCT_MASTER as table name and set **Schema** to INVENTORY, and leave the Tablespace setting as the default for the user, which is INVENTORY.

   e) Enter the first five columns' information (including Data Type, Size, Scale, and Not NULL); but no constraints yet.

   f) Click **Add 5 Table Columns**, and then enter the last column, PRICE, along with its type and size.

   g) Click the **Constraints** tab.

   h) Set the drop-down list value to PRIMARY, and then click **Add**.

   i) Give the constraint a name by entering PK_INV in the **Name** field.

   j) Double-click PRODUCT_ID in the left list to move it to the right list, to make it alone the primary key. Then, click **Continue**.

Solutions for Practice 7: Managing Schema Objects (continued)

k) Set the constraint type drop-down list value to CHECK, and then click **Add**.

l) Enter `CHK_REORDER` for the name of the check constraint.

m) Enter "`reorder_threshold > 0`" (without the quotation marks) in the **Check Condition** field.

> **Definition**
>
> Name `CHK_REORDER`
> * Check
> Condition `reorder_threshold > 0`
>
> **Attributes**
>
> ☐ Disabled
> ☐ Deferrable - In subsequent transactions this allows constraint checking to be deferred until the end of the transaction.
>   ☐ Initially Deferred - Set the default deferred behavior to check constraints at the end of a transaction.
> ☑ Validate - Check to ensure all existing data meets the constraint criteria.
> ☐ Do not enforce the constraint (RELY) - Constraint is not used to enforce data integrity. It is used to express the relationship between tables and views.

n) Leave the Attributes settings at their default settings, and then click **Continue**.

o) Your constraint list should now look like this:

Select	Name	Type	Table Columns	Disabled	Deferrable	Initially Deferred	Validate	RELY
○	PK_INV	PRIMARY	PRODUCT_ID	NO	NO	NO	YES	NO
◉	CHK_REORDER	CHECK		NO	NO	NO	YES	NO

p) Click **OK** to create the table. If you receive errors, correct them, and then click **OK** again.

Solutions for Practice 7: Managing Schema Objects (continued)

3. In the INVENTORY tablespace, create the PRODUCT_ON_HAND table in the INVENTORY schema. You have been given the lab_07_03.sql script to create the table, but there is a problem with it (*intentionally created to enhance your learning experience*). Fix the problem, and run the script. If you cannot find the error right away, then go ahead and run the original script in SQL*Plus to see the error message. This helps you discover and solve the problem. The specifications of the table are:

   PRODUCT_ID number(7). This field should have a foreign key constraint linking it to the PRODUCT_ID field in the PRODUCT_MASTER table.
   QUANTITY number(5)
   WAREHOUSE_CITY varchar2(30)
   LAST_UPDATE date

   a) Edit the lab_07_03.sql in the /home/oracle/labs directory. It has an error in it. If you can spot the error, make the change to correct it. Run the script to create the table by entering this on the OS command line:

   ```
   $ sqlplus dba1/oracle @lab_07_03.sql
   ```

   b) The error in the script is that "(PRODUCT_ID)" is missing after "FOREIGN KEY." So, add "(PRODUCT_ID)".

   ```
   SQL> CREATE TABLE INVENTORY.PRODUCT_ON_HAND
     2  (
     3  PRODUCT_ID NUMBER(7),                      (PRODUCT_ID)
     4  QUANTITY NUMBER(5),
     5  WAREHOUSE_CITY VARCHAR2(30),
     6  LAST_UPDATE DATE,
     7    CONSTRAINT FK_PROD_ON_HAND_PROD_ID
     8      FOREIGN KEY REFERENCES
     9      INVENTORY.PRODUCT_MASTER (PRODUCT_ID) VALIDATE
    10  )
    11  /
       FOREIGN KEY REFERENCES
                    *
   ERROR at line 8:
   ORA-00906: missing left parenthesis
   ```

   c) Remove the prompt commands:

   ```
   prompt   There is an error in this statement. It will not
   prompt   run successfully unless fixed.
   ```

   d) Run the script. The table should be created without error.

Solutions for Practice 7: Managing Schema Objects (continued)

4. In the INVENTORY tablespace, create the OBSOLETE_PRODUCTS table in the INVENTORY schema. This table definition is very much like that of the PRODUCT_MASTER table, so you can use Enterprise Manager's ability to "Define Using SQL" rather than using "Column Specification." The specifications of the table are:

> PRODUCT_ID number(7). This is the primary key field.
> PRODUCT_NAME varchar2(50) with a Not Null constraint
> CODE varchar2(20) with a Not Null constraint
> COST number(5,2)
> PRICE number(5,2)

a) In Enterprise Manager, select **Administration** > Schema > Database Objects > **Tables**.

b) Click **Create**.

c) Leave the default setting for **Table Organization**, and then click **Continue**.

d) Enter OBSOLETE_PRODUCTS for **Name**.

e) Enter INVENTORY for **Schema**.

f) Leave the Tablespace setting as the default for this schema.

g) Set the **Define Using** drop-down list to **SQL**.

h) In the SQL region, enter the following statement:

```
SELECT product_id, product_name, code, cost, price
FROM inventory.product_master
```


i) Click **OK** to create the table.

Solutions for Practice 7: Managing Schema Objects (continued)

5. In the `INVENTORY` tablespace, create an index called `OBS_CODE` on the `CODE` column of the `OBSOLETE_PRODUCTS` table in the `INVENTORY` schema. Choose an appropriate index type: either B-tree or Bitmap. Explain the reason for your choice.

   a) *Question:* Which type of index is appropriate, and why?

   Answer: B-tree, because the `CODE` column can contain many different values, not just a small, finite set.

   b) Select **Administration** > Schema > Database Objects > **Indexes**, and the click **Create**.

   c) Enter `OBS_CODE` as **Name**.

   d) Enter `INVENTORY` as **Schema**.

   e) Enter `INVENTORY.OBSOLETE_PRODUCTS` as **Table Name**.

   f) Click **Populate Columns**. The column names have been filled into the table. If your browser prompts you to remember entered values, click **No**.

   g) Enter an `ORDER` value of `1` for the `CODE` column.

   Table Columns

Column Name	Data Type	Sorting Order	Order
PRODUCT_ID	NUMBER	ASC	
PRODUCT_NAME	VARCHAR2	ASC	
CODE	VARCHAR2	ASC	1
COST	NUMBER	ASC	
PRICE	NUMBER	ASC	

   (Add Column Expression)

   h) Click **Show SQL**, confirm the SQL statement looks like the statement below, and then click **Return**.
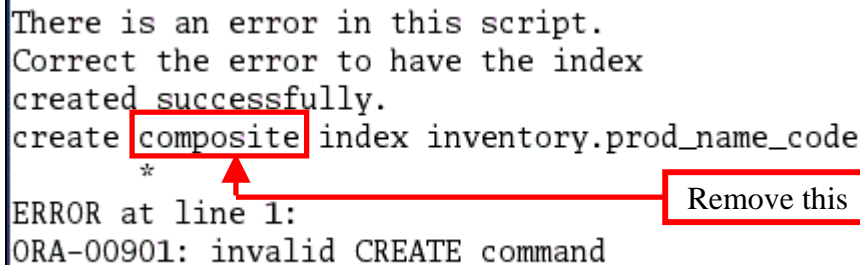
   ```
   CREATE INDEX "INVENTORY"."OBS_CODE" ON
   "INVENTORY"."OBSOLETE_PRODUCTS" ("CODE")
   ```

   i) Click **OK** to create the index.

Solutions for Practice 7: Managing Schema Objects (continued)

6. In the INVENTORY tablespace, create an index called PROD_NAME_CODE on the combined PRODUCT_NAME and CODE columns of the PRODUCT_MASTER table. Use the lab_07_06.sql script *(which contains an error, intentionally created to enhance your learning experience)*. Correct the error and run the script. If you cannot find the error right away, then run the original script in SQL*Plus to see the error message. This will help you discover and solve the problem.

a) Edit the lab_07_06.sql in the /home/oracle/labs directory. It has an error in it. If you can spot the error, make the change to correct it.

```
There is an error in this script.
Correct the error to have the index
created successfully.
create composite index inventory.prod_name_code
        *
ERROR at line 1:
ORA-00901: invalid CREATE command
```
Remove this

The corrected statement is:

```
create index inventory.prod_name_code
on inventory.product_master(product_name,code)
```

b) Remove the prompt commands:

```
prompt   There is an error in this script.
prompt   Correct the error to have the index
prompt   created successfully.
```

c) Run the script to create the index by entering the following on the OS command line:

```
$ sqlplus dba1/oracle @lab_07_06.sql
```

d) The error in the script is that it should not have the word "COMPOSITE". Delete the word, run the script, and the index should be created without error.

7. In the INVENTORY tablespace, use SQL*Plus to create a combined index on the PRODUCT_ID and QUANTITY columns of the PRODUCT_ON_HAND table. The index name should be POH_PROD_ID_QTY.

a) At the SQL*Plus prompt, enter the following command:

```
SQL> create index inventory.poh_prod_id_qty on
inventory.product_on_hand(product_id, quantity);
```

Solutions for Practice 7: Managing Schema Objects (continued)

8. You receive an update for the inventory application that requires you to add two columns to the PRODUCT_MASTER table. Add a column called PRIMARY_SOURCE of the data type varchar2(50). Add another column called SECONDARY_SOURCE of the data type varchar2(50). What is the SQL that executes to do this?

   a) Select **Administration** > Schema > Database Objects > **Tables**.

   b) Enter INVENTORY in the **Schema** field, and then click **Go**.

   c) Select the PRODUCT_MASTER table, and then click **Edit**.

   d) Enter PRIMARY_SOURCE in the **Name** field under PRICE, and set **Data Type** to VARCHAR2, and **Size** to 50.

   e) Enter SECONDARY_SOURCE in the next available **Name** field, and set **Data Type** to VARCHAR2, and **Size** to 50.

   f) Click **Show SQL**.

```
ALTER TABLE "INVENTORY"."PRODUCT_MASTER"
ADD ( "PRIMARY_SOURCE" VARCHAR2(50),
"SECONDARY_SOURCE" VARCHAR2(50))
```

   g) Click **Return**, and then click **Apply**.

9. You receive another update for the inventory application. This change request instructs you to drop the OBSOLETE_PRODUCTS table and add the OBSOLETED column to the PRODUCT_MASTER table, with data type DATE. Do this using EM. What clause is added to the end of the DROP TABLE statement to also remove the table constraints?

   a) Select **Administration** > Schema > Database Objects > **Tables**.

   b) Enter INVENTORY in the **Schema** field, and then click **Go**.

   c) Select the OBSOLETE_PRODUCTS table, and then click **Delete With Options**.

   d) Keep the default settings for performing DROP, and then click **Show SQL** to see the CASCADE CONSTRAINTS option.

```
DROP TABLE "INVENTORY"."OBSOLETE_PRODUCTS" CASCADE CONSTRAINTS
```

   e) Click **Return**, and then click **Yes**.

   f) Select the PRODUCT_MASTER table, and then click **Edit**.

Solutions for Practice 7: Managing Schema Objects (continued)

g) In the first empty **Name** field, enter OBSOLETED, and set **Data Type** to DATE.

h) Click **Apply**.

10. Another change request to the inventory application instructs you to create a view called
    WAREHOUSE_VW (Navigation aid: Administration > Views). The view is in the
    INVENTORY schema and displays (in this order):

    - The name of the product
    - The amount of "product on hand"
    - The warehouse city name

    You will have to join two tables together to create this view.

    a) Select **Administration** > Schema > Database Objects > **Views**.

    b) Click **Create**.

    c) Enter WAREHOUSE_VW in the **Name** field, and INVENTORY in the **Schema** field.

    d) Enter the following in the **Query Text** field:

    ```
    select product_name, quantity, warehouse_city
    from product_master pm, product_on_hand poh
    where pm.product_id = poh.product_id
    ```


    e) Click **OK**.

Solutions for Practice 7: Managing Schema Objects (continued)

11. You receive a notice from developers that there is a certain type of query that will be run very frequently, and they want to be sure that it runs in less than one second. You need to run the query and see how it performs. First, run the `lab_07_11_a.sql` script to generate some test data of the volume indicated in the request. Then, run the query in the `lab_07_11_b.sql` script several times to see the average run time. Note that it takes several seconds to run each time. Create a function-based index on the `CODE` column that will improve the performance of this query.

    a) Right-click the desktop and select **Open Terminal**. Then enter the following at the command prompt to populate the table with test data. This will take three to five minutes to run.

    ```
    $ cd /home/oracle/labs
    $ sqlplus dba1/oracle @lab_07_11_a.sql
    ```

    b) Enter the following on the command line to run the test query.

    ```
    SQL> @lab_07_11_b.sql
    ```

    c) Enter / (a slash), and press [Enter] to run it again. Repeat this several times until you establish an average run time. It should take several seconds each time.

    d) Select **Administration** > Schema > Database Objects > **Indexes**.

    e) Click **Create**.

    f) Enter CODE_FUNC in the **Name** field.

    g) Enter INVENTORY in the **Schema** field.

    h) Enter INVENTORY.PRODUCT_MASTER in the **Table Name** field.

    i) Click **Populate Columns**.

    j) Click **Add Column Expression**.

    k) Enter upper(substr(code,5,2)) in the newly added empty **Column Name** field.

    l) Enter 1 in the **Order** field beside the expression that you just entered.

    m) Click **Show SQL** to confirm that the SQL statement looks like this:

    ```
    CREATE INDEX "INVENTORY"."CODE_FUNC" ON "INVENTORY"."PRODUCT_MASTER"
    (upper(substr(code,5,2)))
    ```

Solutions for Practice 7: Managing Schema Objects (continued)

n) Click **Return**.

o) Click **OK**.

p) Return to the SQL*Plus command line and run the benchmark query a few more times. Note that the execution time is greatly reduced.

12. Use *i*SQL*Plus to identify the data dictionary view name that you would use to list all constraints that the INVENTORY user can see.

a) In the Enterprise Manager browser session, select **Database** > Related Links > **iSQL*Plus**.

The *i*SQL*Plus Connection Role page appears.

b) Select **Normal**, and then click **Continue**.

c) On the Login page, enter inventory as **Username** and verysecure as the **Password**, leave the **Connect Identifier** set to its default value, and then click **Login**.

The Confirm window appears, with the Password Manager offering to remember this login for you. Click **No**.

d) You are looking for things that the INVENTORY user has access to. So, you must use a view with the ALL_ prefix. You can assume that the name of the view that shows constraint information begins with "ALL_CON." In the *i*SQL*Plus Workspace, enter:

```
SELECT * FROM dictionary
WHERE table_name like 'ALL_CON%'
ORDER BY table_name;
```

e) Click the **Execute** button.

Solutions for Practice 7: Managing Schema Objects (continued)

f) Review the returned rows. One of them has the COMMENTS value "Constraints definitions on accessible tables." The object name is ALL_CONSTRAINTS.

13. How many indexes are owned by the INVENTORY user? You are looking for database objects owned by the INVENTORY user, so you know that the data dictionary view begins with the "USER_" prefix. The view name is USER_INDEXES.

a) In the *i*SQL*Plus Workspace, enter:

```
SELECT * FROM user_indexes;
```

b) Click the **Execute** button.

How many indexes are owned by the INVENTORY user?

Answer: 4

Solutions for Practice 7: Managing Schema Objects (continued)

c) Note that there are four rows returned.

d) Log out of *i*SQL*Plus by clicking **Logout** in the top-right corner of the window. Then, close the window by clicking **x** (Close icon) in the top-right corner.

Solutions for Practice 8: Managing Data and Concurrency

Background: The Help desk just received a call from Susan Mavris, an HR representative, complaining that the database is "frozen." Upon questioning the user, you find that she was trying to update John Chen's personnel record with his new phone number, but when she entered the new data, her session froze and she could not do anything else. SQL script files are provided for you in the `/home/oracle/labs` directory.

1. Make an uncommitted update to the row in question by running the `lab_08_01.sql` script. Do not worry if the session seems to "hang"—this is the condition you are trying to create.

   a) Enter the following to run the script. When the script completes executing, you will see a note stating that an uncommitted update has been made.

   ```
   $ sqlplus dba1/oracle @lab_08_01.sql
   ```

   ```
   SQL> show user
   USER is "NGREENBERG"
   SQL> update hr.employees set phone_number='650.555.1212' where employee_id = 110;

   1 row updated.

   SQL> prompt User "ngreenberg" made an update and left it uncommitted in this session.
   User "ngreenberg" made an update and left it uncommitted in this session.
   SQL>
   SQL>
   SQL>
   ```

2. Make an attempt to update the same row in a separate session by running, in a separate terminal window, the `lab_08_02.sql` script. Make sure you see the message "Update is being attempted now" before moving on.

   a) Right-click the desktop and select **Open Terminal** to start another command shell. Then enter the following to run the second script.

   ```
   $ sqlplus dba1/oracle @lab_08_02.sql
   ```

   ```
   Sleeping for 20 seconds to ensure first process gets the lock first.

   PL/SQL procedure successfully completed.

   Sleep is finished.
   Connected.
   USER is "SMAVRIS"
   Update is being attempted now.
   ```

Solutions for Practice 8: Managing Data and Concurrency (continued)

3. Using the **Blocking Sessions** link on the Performance page, detect which session is causing the locking conflict.

   a) In Enterprise Manager, click the Performance page.

   b) The first time you invoke the Performance page, a Software License Agreement is displayed. Press [A] to accept and continue.

   c) Click **Blocking Sessions** in the **Additional Monitoring Links** area.

Blocking Sessions

Page Refreshed **May 31, 2005 5:07:44 PM**

(View Session) (Kill Session)

Expand All | Collapse All

Select	Username	Sessions Blocked	Session ID	Session Serial Number	SQL Hash Value	Wait Class	Wait Event	P1	P2	P3	Seconds in Wait
○	▼ Blocking Sessions										
◉	▼ NGREENBERG	1	133	1784		Idle	SQL*Net message from client	1650815232	1	0	538
○	SMAVRIS	0	158	11895	6smgtv6h8958b	Application	enq: TX - row lock contention	1415053318	327682	540	538

4. What was the last SQL statement that the blocking session executed?

   a) Select the NGREENBERG session, and then click **View Session**.

   b) Click the hash value link named "**Previous SQL**."

Application

Current SQL	**None**
Current SQL Command	**UNKNOWN**
Previous SQL	6smgtv6h8958b
Last Call Elapsed Time	**21 Minutes, 13 Seconds**
SQL Trace	**DISABLED**
Open Cursors	0
Program	**sqlplus@EDRSR9P1 (TNS V1-V3)**
Service	SYS$USERS
Current Module	SQL*Plus
Current Action	**Unavailable**

Solutions for Practice 8: Managing Data and Concurrency (continued)

    c)   Note the SQL that was most recently run.

```
Text
   update hr.employees set phone_number='650.555.1212'
   where employee_id = 110
```

5. Resolve the conflict in favor of the user who complained, by killing the *blocking* session. What SQL statement resolves the conflict?

    a)   Click the browser's **Back** button.

    b)   Now, on the Session Details: NGREENBERG page, click **Kill Session**.

    c)   Leave the Options set to **Kill Immediate**, and then click **Show SQL** to see the statement that is going to be executed to kill the session.

Note: Your session and serial number are most likely to be different from those shown here.

```
ALTER SYSTEM KILL SESSION '133,1784' IMMEDIATE
```

    d)   Click **Return**, and then click **Yes** to carry out the KILL SESSION command.

6. Return to the SQL*Plus command window, and note that SMAVRIS's update has now completed successfully. It may take a few seconds for the success message to appear.

```
USER is "SMAVRIS"
Update is being attempted now.

1 row updated.

Update is completed.
SQL>
```

    a)   Close all open SQL sessions by entering exit, and then close the terminal windows.

Solutions for Practice 9: Managing Undo Data

Background: A new version of your application will include several reports based on very long-running queries. Configure your system to support these reports.

1. Use the Undo Advisor to calculate the amount of undo space required to support a report that takes two days to run, on the basis of an analysis period of the last seven days.

   a) In Enterprise Manager, select **Administration** > Related Links > **Advisor Central**.

   b) Click **Undo Management**.

   c) Click **Undo Advisor**.

   d) In the **New Undo Retention** field, enter 2. Then select **days** from the drop-down list.

   e) Select Last Seven Days in the **Analysis Time Period** drop-down list. Results of the analysis will be displayed.
   Note: The values that you see are likely to be different from those shown here.

   f) *Question:* What does the analysis recommend as "Required Tablespace Size for New Undo Retention"?

   Answer: 110 MB

   g) Click **OK**.

Solutions for Practice 9: Managing Undo Data (continued)

2. Resize the undo tablespace to support the retention period required by the new reports (or 1 GB, whichever is smaller). Do this by increasing the size of the existing data file.

    a) Click **Edit Undo Tablespace**.

    b) *Question:* What are the two ways to add space to a tablespace?

    Answer: Add a new data file or increase the size of an existing data file.

    c) To increase the size of the existing data file, select the data file, and then click **Edit**.

    d) Enter the new value from step 1 (f) in the **File Size** field. Round off to the nearest 100 MB (not to exceed 1 GB). Make sure the box for "Automatically extend datafile when full" is <u>not</u> selected. Then, click **Continue**.

    e) Click **Show SQL** to confirm that the SQL statement looks similar to this:

```
ALTER DATABASE DATAFILE
'/u01/app/oracle/oradata/ORCL/datafile/o1_mf_undotbs1_1pzfonlk_.dbf'
RESIZE 200M
ALTER DATABASE DATAFILE
'/u01/app/oracle/oradata/ORCL/datafile/o1_mf_undotbs1_1pzfonlk_.dbf'
AUTOEXTEND OFF
```

    f) Click **Return**, then click **Apply** to implement your change.

Solutions for Practice 10: Implementing Oracle Database Security

Background: You have just been informed of suspicious activities in the HR.JOBS table in your ORCL database. All maximum salaries seem to fluctuate in a strange way. You decide to enable standard database auditing and monitor data manipulation language (DML) activities in this table.

Log in as the DBA1 user (with oracle password, connect as SYSDBA) and perform the necessary tasks either through Enterprise Manager Database Control or through SQL*Plus. All scripts for this practice are in the /home/oracle/labs directory.

1. Use Enterprise Manager to enable database auditing. Set the AUDIT_TRAIL parameter to XML.

   a) Invoke Enterprise Manager as the DBA1 user in the SYSDBA role for your ORCL database.

   b) Select **Administration** > Schema > Users & Privileges > **Audit Settings**.

   c) Click the **NONE** link.

   d) On the Initialization Parameters page, click the **SPFile** tab.

   e) Enter audit in the **Name** field and then click **Go**.

   f) For the **audit_trail** parameter, select the **XML** value.

   g) Click **Show SQL**.

   h) Review the statement and then click **Return**.

   i) On the Initialization Parameters page, click **Apply**.

Solutions for Practice 10: Implementing Oracle Database Security (continued)

2. Because you changed a static parameter, you must restart the database. Do so by running the `lab_10_02.sh` script.

   a) In a terminal window, enter:

   ```
   cd /home/oracle/labs
   ./lab_10_02.sh
   ```

   Continue with the next step when you see that the database is restarted and the script has exited out of SQL*Plus.

3. Back in Enterprise Manager, select `HR.JOBS` as the audited object and `DELETE`, `INSERT`, and `UPDATE` as **Selected Statements**. Gather audit information by session.

   a) Click the Database home page tab to ensure that Enterprise Manager had time to update the status of the database and its agent connections. Because the database has been restarted, you have to log in to EM again as the `DBA1` user.

   b) Then, select **Administration** > Schema > Users & Privileges > **Audit Settings**.

   c) Click the **Audited Objects** tab, and then click the **Add** button.

   d) On the Add Audited Object page, ensure that the **Object Type** is Table, and enter `HR.JOBS` in the **Table** field (or use the flashlight icon to retrieve this table).

   e) Move `DELETE`, `INSERT`, and `UPDATE` into the **Selected Statements** area by double-clicking each of them.

   f) Click **Show SQL**.

Solutions for Practice 10: Implementing Oracle Database Security (continued)

    g) Review the statement, and then click **Return**.

    h) Click **OK** to activate this audit.

4. Provide input for the audit, by executing the `lab_10_04.sh` script. This script creates the `AUDIT_USER` user, connects to SQL*Plus as this user, and multiplies the values in the `MAX_SALARY` column by `10`. Then, the `HR` user connects and divides the column values by `10`. Finally, the `AUDIT_USER` user is dropped again.

    a) In a terminal window, enter:

```
cd /home/oracle/labs
./lab_10_04.sh
```

5. In Enterprise Manager, review the audited objects.

    a) Select **Administration** > Schema > Users & Privileges > **Audit Settings**.

    b) Click **Audited Objects** in the **Audit Trails** area, which is on the right side of the page.

    c) On the Audited Objects page, review the collected information, and optionally click **Show SQL**.

    d) Click **Return**.

6. Undo your audit settings for `HR.JOBS`, disable database auditing, and then restart the database by using the `lab_10_06.sh` script.

    a) On the Audit Settings page, click the **Audited Objects** tab.

    b) Enter `HR` as **Schema,** and then click **Search**.

    c) Select all three rows, and then click **Remove**.

Solutions for Practice 10: Implementing Oracle Database Security (continued)

    d) On the Confirmation page, click **Show SQL**.

> ![Confirmation]
>
> **Confirmation**
>
>   (No) (Yes)
>
> Are you sure you want to remove the 3 selected audited objects?
>
> The audited statements you remove will no longer be audited on the objects.
>
> ▼Hide SQL
> ```
> NOAUDIT DELETE ON HR.JOBS
> NOAUDIT INSERT ON HR.JOBS
> NOAUDIT UPDATE ON HR.JOBS
> ```

    e) Review the statements, then click **Yes** to confirm your removal.

    f) On the Audit Settings page, click **XML** in the **Configuration** region.

    g) On the Initialization Parameters page, click the **SPFile** tab.

    h) On the SPFile page, enter `audit` in the **Name** field, and then click **Go**.

    i) For the **audit_trail** parameter, select the **NONE** value.

    j) Click **Show SQL**.

> **Show SQL**
>
> ```
> ALTER SYSTEM SET audit_trail = "NONE" SCOPE=SPFILE
> ```

    k) Review the statement, and then click **Return**.

    l) On the Initialization Parameters page, click **Apply**.

    m) Because you changed a static parameter, you must restart the database. Do so by running the `lab_10_06.sh` script. In a terminal window, enter:

```
cd /home/oracle/labs
./lab_10_06.sh
```

7. Maintain your audit trail: Because you are completely finished with this task, delete all audit files from the `/u01/app/oracle/admin/orcl/adump` directory.

    a) In a terminal window, enter:

```
cd /u01/app/oracle/admin/orcl/adump
ls
rm -f *
```

    b) Close the terminal window.

Solutions for Practice 11: Configuring the Oracle Network Environment

Background: Users need to connect to your `ORCL` database. Work with them to enable connections by using different methods. Ensure that users can use connect-time failover to take advantage of a backup listener.
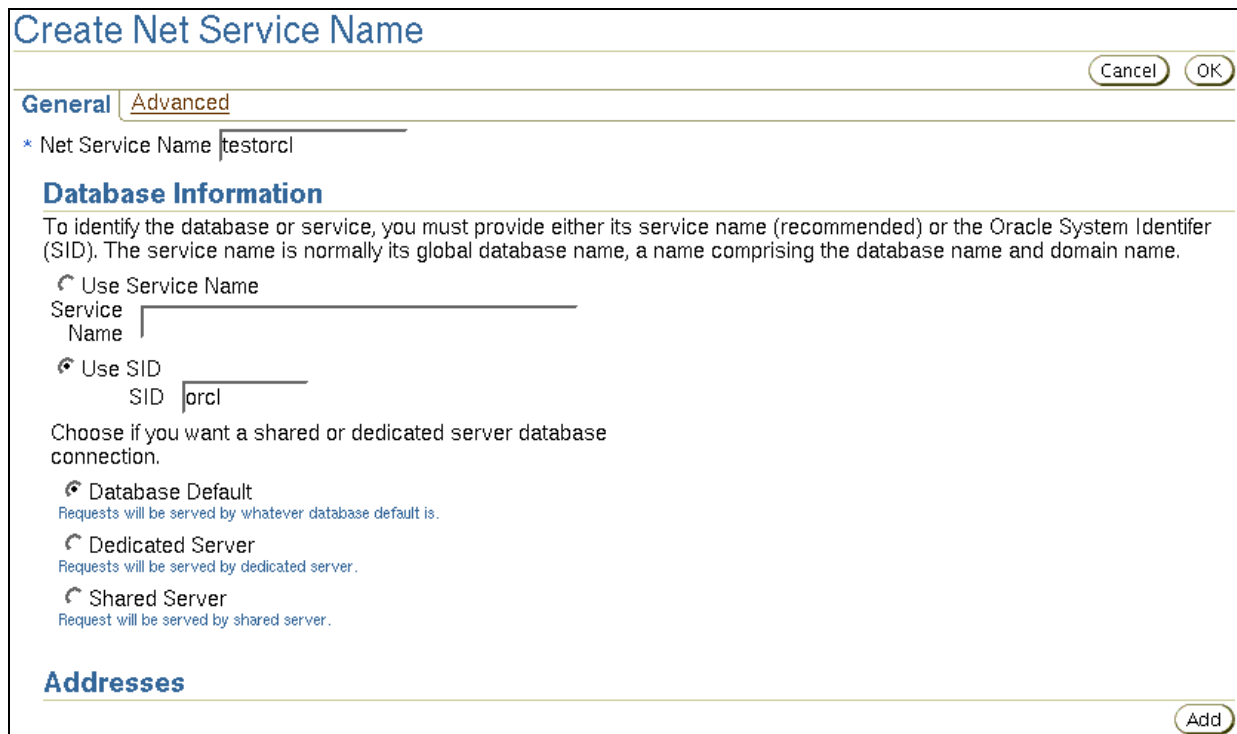
1. Make a copy of your `listener.ora` and `tnsnames.ora` files. They are in the `$ORACLE_HOME/network/admin` directory.

    a) In an terminal window, enter `cd $ORACLE_HOME/network/admin` to navigate to the `/u01/app/oracle/product/10.2.0/db_1/network/admin` directory.

    b) Enter `cp listener.ora listener.old` to create a copy of the `listener.ora` file.

    c) Enter `cp tnsnames.ora tnsnames.old` to create a copy of the `tnsnames.ora` file.

    d) Enter `ls -l`, if you want to see the copies and their privileges in your directory.

2. Navigate to the Net Services Administration page. Start by clicking the **Listener** link on the Database home page.

    a) Invoke Enterprise Manager as the `DBA1` user in the `SYSDBA` role for your `ORCL` database.

    b) On the Database Instance – Home page, click the **Listener** link in the **General** region.

    c) In the **Related Links** region, click **Net Services Administration**.

3. Modify your local Names Resolution file so that you can connect to another database. Name the connection to a partner's `ORCL` database **testorcl**.

    a) On the Net Services Administration page, select **Local Naming** from the Administer drop-down list, and then click **Go**.

    The **Netservices Administration: Host Login** page appears.

    b) If you previously saved the `oracle` username and `oracle` password as preferred credentials for your host login, then they appear on the screen. If not, enter **oracle** as Username and Password, select the **Save as Preferred Credential** check box, and then click **Login**.

    c) On the Local Naming page, click **Create** to enter a new network service name.

    d) Enter **testorcl** as Net Service Name.

    e) Select **Use SID**, and enter **orcl** as SID.

Solutions for Practice 11: Configuring the Oracle Network Environment (continued)

    f)   Select **Database Default**.

    g)   Click **Add** in the **Addresses** region.

Click **No** if you are asked, "**Do you want to remember the values you filled in?**"

    h)   On the Add Address page, select the following values:

       - Protocol: **TCP/IP**
       - Port: **1521**
       - Host: <*Enter the host name or IP address of **your partner's computer**. It could be something like* edrsr9p1.us.oracle.com *or like* 139.185.35.109>

Solutions for Practice 11: Configuring the Oracle Network Environment (continued)

    i)   Click **OK** to return to the Create Net Service Name properties page.

    j)   Click **OK**.

    The Creation Message appears: Net Service "testorcl" created successfully.

4.   In Enterprise Manager, test access to your partner's ORCL database as the **system** user with the **oracle** password by using the **testorcl** Local Naming.

    a)   Select **testorcl** on the Local Naming page, and then click **Test Connection**.

    The "Test Connection To Net Service Name: testorcl" appears.

    b)   Enter **system** as Username and **oracle** as Password, and then click **Test**.

    The Processing page displays status information. Then, it is followed by a success message. *If you receive any errors or warnings, resolve them.*

5.   Test your changes to the network configuration by using SQL*Plus or *i*SQL*Plus. Again, use: system/oracle@testorcl. To see your partner's information, select the instance_name and host_name columns from the v$instance table.

    a)   In an terminal window, enter:

```
sqlplus system/oracle@testorcl
```

    The Oracle SQL*Plus window opens. *If you receive any errors or warnings, resolve them.*

    b)   At the SQL> prompt, enter the following command:

```
select instance_name, host_name from v$instance;
```

Solutions for Practice 11: Configuring the Oracle Network Environment (continued)

```
[oracle@EDRSR3OP1 oracle]$ sqlplus system/oracle@testorcl

SQL*Plus: Release 10.2.0.0.0 - Beta on Mon May 16 16:38:46 2005

Copyright (c) 1982, 2004, Oracle.  All rights reserved.


Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.0.0 - Beta
With the Partitioning, OLAP and Data Mining options

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
----------------
HOST_NAME
------------------------------------------------------------
orcl
EDRSR9P1


SQL>
```

You should see your partner's host name.

   c) Enter **exit** to exit your SQL*Plus session.

6. Create a LISTENER2 listener to support connect-time failover. Use port 1561 for this listener. First, log out of Enterprise Manager and run the lab_11_06.sh script to configure the NetProperties file.

   a) Logout of Enterprise Manager.

   b) Run the lab_11_06.sh script at the operating system command prompt.

   c) Log in to Enterprise Manager as the DBA1 user in the SYSDBA role for your ORCL database.

   d) On the Database Instance – Home page, click the **Listener** link in the **General** region.

   e) In the **Related Links** region, click **Net Services Administration**.

   f) On the Net Services Administration page, select **Listeners** from the **Administer** drop-down list, and then click **Go**. Enter host credentials as oracle and oracle for username and password, and then click **Login**.

Solutions for Practice 11: Configuring the Oracle Network Environment (continued)

g) On the Listeners page, which gives you an overview of the existing listeners, click the **Create** button.

The Create Listener page appears.

h) Enter **LISTENER2** as Listener Name, and then click **Add** to add a listener address.

i) Enter or confirm the following values:

- Protocol: TCP/IP
- Port: **1561**
- Host: <Your computer's host name; for example, edrsr30p1.us.oracle.com>

j) Click **OK**.

k) Click the **Static Database Registration** tab, and then click the **Add** button to connect the new listener with your ORCL database.

l) Enter the following values:

- Service Name: **orcl**
- Oracle Home Directory: **/u01/app/oracle/product/10.2.0/db_1**
- Oracle System Identifier (SID): **orcl**

m) Click **OK** to add the database service.

n) Click **OK** to create the LISTENER2 listener.

Solutions for Practice 11: Configuring the Oracle Network Environment (continued)

7. Start the `LISTENER2` listener.

    a) Confirm that the **LISTENER2** listener and **Start/Stop** Actions are selected, and then click **Go**.

    b) Click **OK** on the Start/Stop page.

    A confirmation message appears with a **View Details** link.

    c) Optionally, click the **View Details** link, review the listener status information, and use the **Back** icon of your browser to return to the previous page.

Solutions for Practice 12: Proactive Maintenance

Background: You want to proactively monitor your ORCL database so that common problems can be fixed before they affect users. This practice session invents some issues so that you can familiarize yourself with the tools that are available. First, execute scripts to set up your Automatic Database Diagnostic Management (ADDM) environment.

1. Create a new, locally managed tablespace called TBSADDM. Its addm1.dbf data file is 50 MB. Ensure that the TBSADDM tablespace does not use Automatic Segment Space Management (ASSM). Execute the lab_12_01.sh script to perform these tasks.

   a) In a terminal window, enter:

   ```
   cd /home/oracle/labs
   ./lab_12_01.sh
   ```

2. Create a new ADDM user, identifed by ADDM. Assign the TBSADDM tablespace as default tablespace. Assign the TEMP tablespace as temporary tablespace. Grant the following roles to the ADDM user: CONNECT, RESOURCE, and DBA. Execute the lab_12_02.sh script to perform these tasks.

   a) In a terminal window, enter:

   ```
   ./lab_12_02.sh
   ```

3. Use the DBMS_ADVISOR package to set the database activity time to 30 minutes. As an ADDM user, drop and create the ADDM table and gather statistics for this table. Create a snapshot in Automatic Workload Repository (AWR). Execute the lab_12_03.sh script to perform these tasks.

   a) In a terminal window, enter:

   ```
   ./lab_12_03.sh
   ```

4. Create activity to be analyzed. Execute the lab_12_04.sh script to perform these tasks.

   a) In a terminal window, enter the following. You may have to press [Enter] after you see that eight PL/SQL procedures have completed, in order to see the command prompt again.

   ```
   ./lab_12_04.sh
   ```

Solutions for Practice 12: Proactive Maintenance (continued)

5. In Enterprise Manager, review the Performance page as a user connected as SYSDBA. View performance data in real time with a 15-seconds refresh cycle. After a while, you should see a spike on the "Average Active Sessions" graph. This is your activity to be analyzed. Looking at the graph, you can already determine that this instance is suffering from concurrency problems.
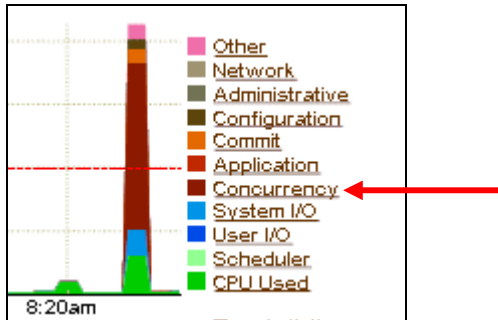
Note: Depending on when you run the workload, you may see differences between your graph and the one provided as a possible solution.

After the spike is finished, execute the lab_12_05.sh script. This script forces the creation of a new snapshot and gathers statistics on your ADDM table.

a) Invoke Enterprise Manager as the DBA1 user in the SYSDBA role for your ORCL database.

b) Click the **Performance** tab.

    If this is the first time that you accessed the Performance page, you need to accept the Adobe license agreement. Follow the directions in the pop-up window to accept the agreement.

Solutions for Practice 12: Proactive Maintenance (continued)
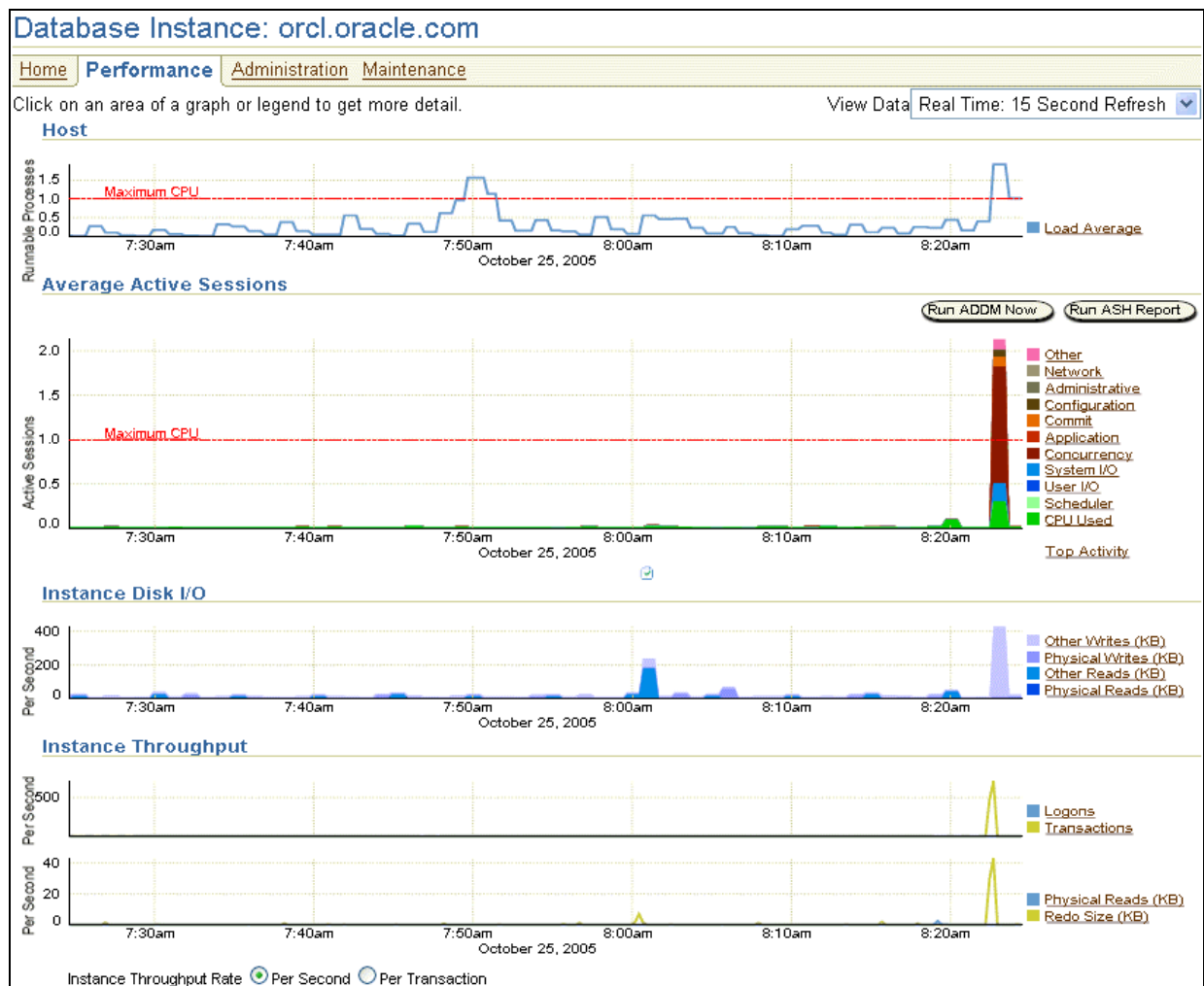
c) After the spike has finished, in a terminal window, enter:

```
./lab_12_05.sh
```
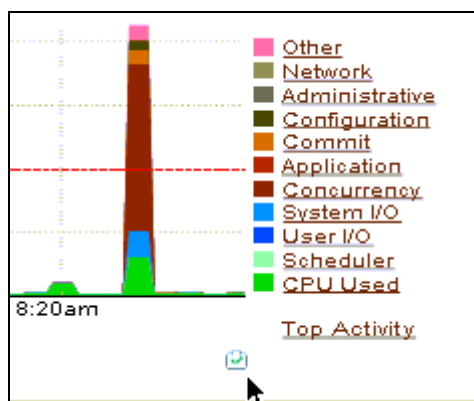

Note: The icon next to the cursor in this screenshot is a shortcut to the Automatic Database Diagnostic Monitor (ADDM) page.

Solutions for Practice 12: Proactive Maintenance (continued)

6. Look at the **Performance Analysis** findings in order of their impact. There are several access paths to this information.

   Looking at the Performance Analysis section, you see that the first finding (in the SQL Tuning Recommendations category) has a 100% impact on the system. So your first impluse is to look at this finding in more detail. However, looking at this SQL statement does not yet help you to understand the concurrency problem of your database.

   Research the next finding under Schema Recommendations: **Read and write contention of database blocks was consuming significant database time**. Here you are advised to use the Automatic Segment Space Management (ASSM) feature for your ADDM table.

   a) Navigate to the Database home page, and then click **Advisor Central** at the bottom of the page.

   b) Your ADDM task should already be displayed. If not, search for it and display it on this page.

   c) Select the task, and then click the **View Result** button (or alternatively, click the name of the task).

Solutions for Practice 12: Proactive Maintenance (continued)

Looking at the Performance Analysis section, you see that the first finding has a 100% impact on the system. So your first impluse is to look at this finding in more detail.

d) Click the **SQL statements consuming significant database time were found** link.

Solutions for Practice 12: Proactive Maintenance (continued)

    e) Review the recommendations on the Performance Finding Details page. However, looking at this SQL statement does not yet help you to understand the concurrency problem of your database. Click the **Back** icon in your Web browser.

    f) On the Automatic Database Diagnostic Monitor (ADDM) page, click the **Read and write contention of database blocks was consuming significant database time** link . This finding appears as type **Schema** under the **Recommendations** heading.

    g) You are advised to use the Automatic Segment Space Management feature for your `ADDM` table.

7. To implement the recommendation, you must re-create the object. Create a new, locally managed tablespace, called `TBSADDM2` with a 50 MB datafile, called `addm2_1.dbf`. Ensure that the `TBSADDM2` tablespace uses the the Automatic Segment Space Management feature. Then, execute the `lab_12_07.sh` script to drop the `ADDM` table, to re-create it in the new tablespace, to gather statistics and to take a new snapshot.

    a) In Enterprise Manager, select **Administration** > Database Administration > Storage > **Tablespaces**.

    b) Click **Create**.

    c) Enter `TBSADDM2` as the tablespace name, and verify that **Extent Management** is **Locally Managed**, **Type** is **Permanent**, **Status** is **Read Write**, and **Use Bigfile tablespace** is <u>not </u>selected.

Solutions for Practice 12: Proactive Maintenance (continued)

    d) Click **Add** in the **Datafiles** region.

    e) On the Add Datafile page, enter `addm2_1.dbf` for **File Name**, and `50 MB` as **File Size**.

    f) Click **Continue**.

    g) Click the **Storage** tab, and verify that **Extent Allocation** is **Automatic**, **Segment Space Management** is **Automatic**, and **Logging** is enabled.

    h) Click the **General** tab.

    i) Click **Show SQL** to see the SQL that will be run, and then click **Return**.

    j) Click **OK**, and a successful Update Message appears.

    k) In a terminal window, enter:

```
./lab_12_07.sh
```

8. Execute your workload again. (The `lab_12_08.sh` script is identical to the `lab_12_04.sh` script.)

    a) In a terminal window, enter the following. You may have to press [Enter] after you see that eight PL/SQL procedures have completed, in order to see the command prompt again.

```
./lab_12_08.sh
```

9. In Enterprise Manager, review the Performance page as a user connected as `SYSDBA`. View performance data in real time with a 15-seconds refresh cycle. After a while, you should see a spike on the "Average Active Sessions" graph.

Note: Depending on when you run the workload, you may see differences between your graph and the one provided as a possible solution.

After the spike is finished, execute the `lab_12_09.sh` script. (The `lab_12_09.sh` script is identical to the `lab_12_05.sh` script.) This script forces the creation of a new snapshot and gathers statistics on your `ADDM` table.

    a) Invoke Enterprise Manager as the `DBA1` user in the `SYSDBA` role for your `ORCL` database.

    b) Click on the **Performance** tabbed page.

Solutions for Practice 12: Proactive Maintenance (continued)
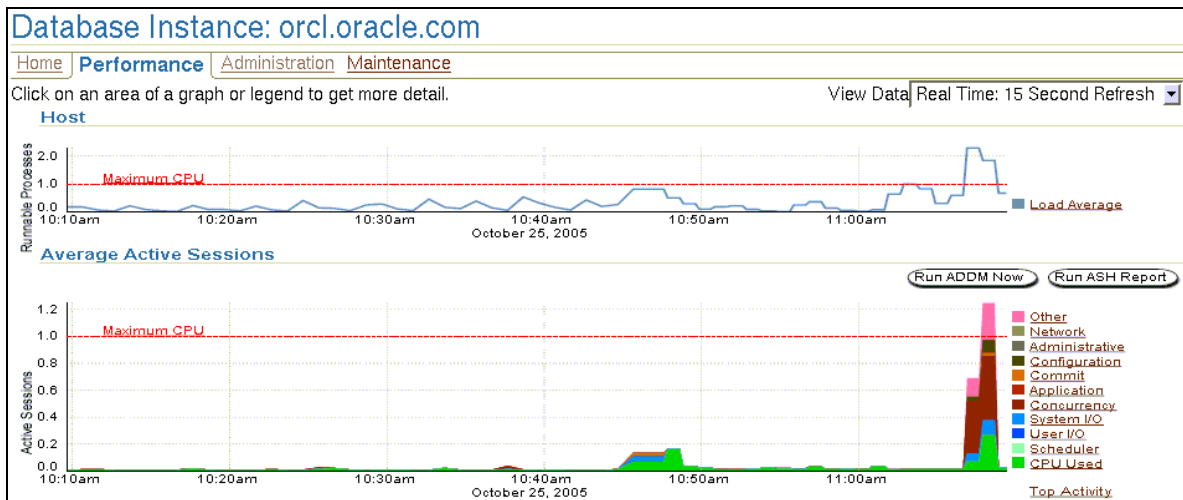
c) After the spike is finished, enter the following in a terminal window:

```
./lab_12_09.sh
```

10. Review the Performance Analysis on the Database home page.

   a) Navigate to the Database home page, and then click **Advisor Central** at the bottom of the page.

   b) Click the topmost ADDM task name.

You see that there are no longer any schema-related recommendations. By moving the ADDM table to the locally managed TBSADDM2 tablespace, which uses the Automatic Autoextend Segment feature, you obviously fixed the root cause of this problem.

Solutions for Practice 12: Proactive Maintenance (continued)

11. To not affect other practice session, execute the `lab_12_11.sh` script to clean up your environment.

    a) In a terminal window, enter:

```
./lab_12_11.sh
```

Solutions for Practice 13: Performance Management

Background: Users are complaining about slower-than-normal performance for operations involving the human resources and order-entry applications. When you question other members of the DBA staff, you find that maintenance was recently performed on some of the tables belonging to the HR schema. You need to troubleshoot and make changes as appropriate to resolve the performance problems. SQL script files are provided for you in the /home/oracle/labs directory. Other directories are individually named.

1. Log in to SQL*Plus as the DBA1 user and perform maintenance on tables in the HR schema by running the lab_13_01.sql script.

    a) Connect to the database as the DBA1 user and perform some maintenance on the employees table by running the lab_13_01.sql script.

```
$ cd labs
$ sqlplus dba1/oracle
SQL> @lab_13_01.sql
```

2. You get calls from HR application users saying that a particular query is taking longer than normal to execute. The query is in the lab_13_02.sql script. As the HR user, run it.

    a) Enter the following in SQL*Plus , while logged in as the HR user:

```
SQL> @lab_13_02.sql
```

3. Using Enterprise Manager, locate the HR session in which the above statement was just executed, and view the execution plan for that statement.

    a) In Enterprise Manager, click **Search Sessions** on the Performance tabbed page.

Solutions for Practice 13: Performance Management (continued)

b) Change the Filter for the search criteria to "DB User," enter HR in the field to the right of that, and then click **Go**.

c) Click the **SID** number in the **Results** listing.

d) You now see the Session Details page for this session. Click the hash value link to the right of the **Previous SQL** label.

Solutions for Practice 13: Performance Management (continued)

e) That shows you the SQL Details page for the last SQL statement executed by that session, which is the one in question. Click the **Plan** tab to see the execution plan for the query.

f) You see in the Operation column that this query is doing a full table scan (TABLE ACCESS FULL). Because you know that the query's condition is an equality comparison on the primary key (EMPLOYEE_ID), you decide to investigate the status of the primary key index.

4. Using Enterprise Manager, check to see the status of the EMPLOYEE table's index on EMPLOYEE_ID. See if it is VALID.

   a) Select **Administration** > Schema > Database Objects > **Indexes**.

   b) Select Table Name as the **Search By** value.

   c) Enter HR as **Schema Name**.

   d) Enter employees for **Object Name**.

   e) Click **Go**, and the list of six indexes appears.

   f) Click the index named EMP_EMP_ID_PK.

   g) Click the index link and note that the status of the index is UNUSABLE.

Solutions for Practice 13: Performance Management (continued)

5. Now that you have seen one index with a non-VALID status, you decide to check all indexes. Using SQL*Plus, as the HR user, find out which HR schema indexes do not have STATUS of VALID. To do this, you can query a data dictionary view with a condition on the STATUS column.

   a) Go to the SQL*Plus session where you are still logged in as the HR user, and run this query:

   ```
   SQL> select index_name, table_name, status
        from user_indexes where status <> 'VALID';
   ```

   b) See that the output lists six indexes, all on the EMPLOYEES table.

   ```
   SQL> select index_name, table_name, status
     2  from user_indexes where status <> 'VALID';

   INDEX_NAME                      TABLE_NAME                      STATUS
   ------------------------------- ------------------------------- --------
   EMP_EMAIL_UK                    EMPLOYEES                       UNUSABLE
   EMP_EMP_ID_PK                   EMPLOYEES                       UNUSABLE
   EMP_DEPARTMENT_IX               EMPLOYEES                       UNUSABLE
   EMP_JOB_IX                      EMPLOYEES                       UNUSABLE
   EMP_MANAGER_IX                  EMPLOYEES                       UNUSABLE
   EMP_NAME_IX                     EMPLOYEES                       UNUSABLE

   6 rows selected.

   SQL>
   ```

6. Using Enterprise Manager, reorganize all the indexes in the HR schema that are marked as UNUSABLE.

   a) In Enterprise Manager, on the page displaying the EMP_EMP_ID_PK index, select **Reorganize** in the **Actions** list, and then click **Go**.

   b) Click **Add**, to add each of the other five indexes to the reorganization operation.

   c) In the Add screen, choose Indexes for the **Type** drop-down list, and enter hr in the **Schema** field.

   d) Click **Search**.

Solutions for Practice 13: Performance Management (continued)

e) Select the five other indexes whose names start with "EMP_."

```
Search
                 Type    Indexes         ▼
              Schema    hr
         Object Name
       Partition Name
          Tablespace
                      ( Search )

Available Objects: Indexes
Select All | Select None
┌──────────────────────────────────────────┐
│ Select│ Name △                            │
├──────────────────────────────────────────┤
│   □   │ HR.DEPT_ID_PK                     │
│   □   │ HR.DEPT_LOCATION_IX               │
│   ☑   │ HR.EMP_DEPARTMENT_IX              │
│   ☑   │ HR.EMP_EMAIL_UK                   │
│   ☑   │ HR.EMP_JOB_IX                     │
│   ☑   │ HR.EMP_MANAGER_IX                 │
│   ☑   │ HR.EMP_NAME_IX                    │
│   □   │ HR.JHIST_DEPARTMENT_IX            │
│   □   │ HR.JHIST_EMPLOYEE_IX              │
│   □   │ HR.JHIST_EMP_ID_ST_DATE_PK        │
└──────────────────────────────────────────┘
```

f) Click **OK**.

g) Click **Next**.

h) Keep all the default settings for **Options**, and then click **Next**.

i) Note that there are no problems reported on **Impact Report**, and then click **Next**.

j) On the Schedule page, enter `oracle` and `oracle` for Username and Password under **Host Credentials**.

k) Click **Next**.

l) On the Review page, click **Submit Job**.

Solutions for Practice 13: Performance Management (continued)

m) After the Confirmation page appears, click the job name to see its status. Click **Reload** on your browser until you see the job has succeeded.

7. Return to the SQL*Plus session where the HR user is logged in, and run the `lab_13_07.sql` script to execute the same kind of query. Then, repeat the steps to see the plan of the last SQL statement executed by this session, to see if the plan has changed.

a) Enter the following at the SQL*Plus prompt:

```
SQL> @lab_13_07.sql
```

b) Repeat the tasks listed in step 3. Note that the plan now uses the index.

c) Quit the SQL*Plus session.

8. What is the difference in execution plans, and why?

Answer: The statement execution uses a unique index scan instead of a full table scan, because the index is usable after your index reorganization.

Solutions for Practice 13: Performance Management (continued)

9. Simulate a working load on your instance by running the `lab_13_09.sql` script as the `DBA1` user. Please note the SID value for task 10.

Possible answer: 147   (Your answer is most likely a different one).

This script takes about 20 minutes to complete. So, run it in a separate terminal window and continue with this practice exercise while it runs.

Note: Because this script generates a fairly heavy load in terms of CPU and disk I/O, you will notice that response time for Database Control is slower.

```
$ sqlplus dba1/oracle
SQL> @lab_13_09.sql
```

    a) In Enterprise Manager, navigate to the Performance page, and investigate system performance.

    b) You may need to wait a minute or two to see the effects of the load generation script appear on the graphs.
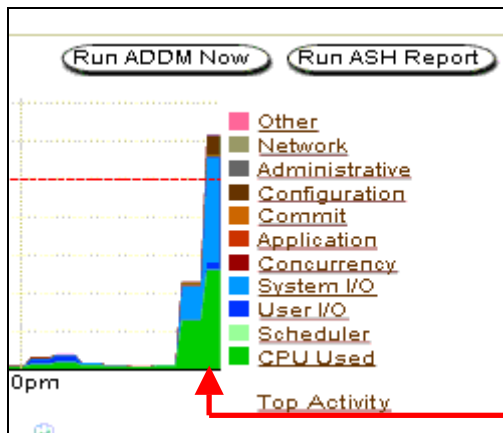
Wait to see the beginning of a spike in the **Average Active Sessions** graph before proceeding.

Question 1: In the **Average Active Sessions** graph, which are the two main categories that active sessions are waiting for?

Answer: System I/O and CPU Used

Solutions for Practice 13: Performance Management (continued)

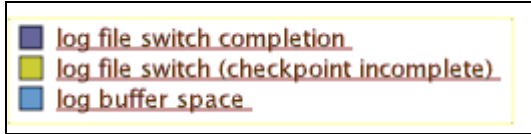*Question 2:* In the Configuration category of waits, what is one of the contributors to the wait time? Click Configuration to see the graph.

Answer: Any one of these:

Question 3: Click Back, and then click Physical Writes on the Instance Disk I/O graph. Determine which process is doing the most writing to the disk.

Answer: DBW0
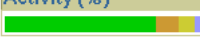
c) Click **Back**.

d) Click **Top Activity** in the **Additional Monitoring Links** region.

e) Click the **SQL ID** of the first SQL statement listed in the **Top SQL** region.

f) See the first SQL statement.

Question 4: What SQL statement is causing the most waits?

Answer: delete from sh.sales_copy

10. Kill the session that is generating the load. Use the session ID recorded in step 9. The session ID should be listed at the top of the list on the right side of the page, under the **Top Sessions** region.

a) Click the SID number for the session ID recorded earlier. This is found under the heading **Detail for Selected 5 Minute Interval.**

Solutions for Practice 13: Performance Management (continued)

b) On the Session Details page, click **Kill Session**, and then click **Yes** to confirm.

Note: If you remain on this Session Details page long enough for a few automatic refreshes to be done, you may see the following warning, which means you are attempting to refresh information about a session that's already been killed. You can ignore this warning.
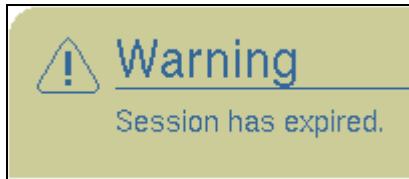
c) Click **Top Activity** in the navigation history at the top of the page. Note that the session activity in the database has declined considerably.
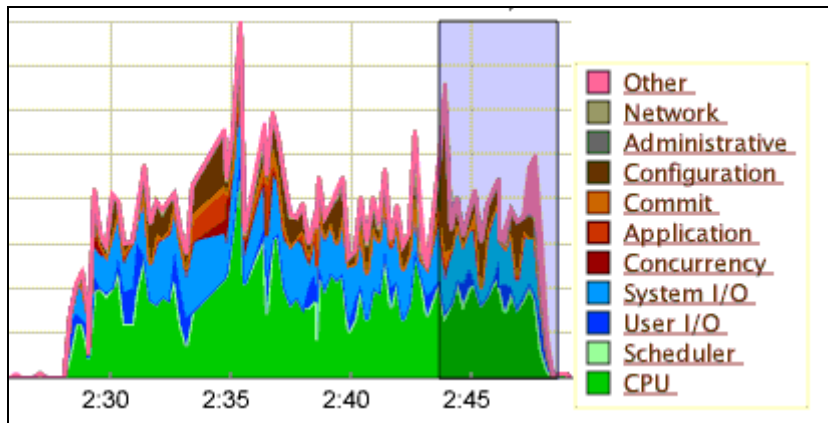
Solutions for Practice 14: Backup and Recovery Concepts

Background: Your ORCL database is ready to move from test or development into production. Configure your database to reduce the chances of failure or data loss.

1. Verify that you have two control files to ensure redundancy.

   a) Invoke Enterprise Manager as the DBA1 user in the SYSDBA role for your ORCL database.

   b) Select **Administration** > Database Administration > Storage > **Control Files**.

   c) *Question 1:* How would you add another control file if you needed to?

   Answer: Adding a control file is a manual operation. To perform this, you must:

   - Shut down the database

   - Use the operating system to copy an existing control file to the location where you want your new file to be

   - Start the database by using Enterprise Manager. Unlike a normal startup, you would use Advanced Options to select a different startup mode. Select "Start the instance" to leave the instance in the NOMOUNT state.

   - Edit the CONTROL_FILES initialization parameter to point to the new control file

   - Continue the STARTUP database operation until the database is in an open state

   Note: This answer does not apply to an OMF database, as the control files in that case would have to all be recreated.

2. Check how many members each redo log group has. Ensure that there are at least two redo log members in each group. In what directory or directories are the redo log files stored?

Solutions for Practice 14: Backup and Recovery Concepts (continued)

a) Select **Administration** > Database Administration > Storage > **Redo Log Groups**, and
note how many members are in the "# of Members" column. There should be two per
group.

Selection Mode	Single ▾						Create
			Edit	View	Delete	Actions Clear logfile ▾	Go

Select	Group △	Status	# of Members	Archived	Size (KB)	Sequence	First Change#
◉	1	Inactive	2	No	51200	26	1408043
○	2	Inactive	2	No	51200	27	1424729
○	3	Current	2	No	51200	28	1452632

b) Select the first group, and then click **Edit** to see the member file names. Note that one
member is in directories under the `oradata` directory, and another is in the Flash
Recovery Area. Click the browser's **Back** button.

Redo Log Members

		Add
		Edit Remove

Select	File Name	File Directory
◉	o1_mf_1_1p4sbq1m_.log	/u01/app/oracle/oradata/ORCL/onlinelog/
○	o1_mf_1_1p4sbs8b_.log	/u01/app/oracle/flash_recovery_area/ORCL/onlinelog/

Note: In a production database, you want to ensure that the two members are on different
hard drives, preferably with different disk controllers, to minimize the risk of any single
hardware failure destroying an entire log group.

3. You notice that, for each log group, the `Archived` column has a value of `No`. This means
that your database is not retaining copies of redo logs to use for database recovery, and in
the event of a failure, you will lose all data since your last backup. Place your database in
`ARCHIVELOG` mode, so that redo logs will be archived.

a) Create a new directory `/u01/app/oracle/archive` as the destination for the redo
log files.

- In a terminal window, enter **cd /u01/app/oracle**.

- Then, enter **mkdir archive**.

- Optionally, enter **ls -l** to view your new directory and its OS permissions.

Solutions for Practice 14: Backup and Recovery Concepts (continued)

b) In Enterprise Manager, select **Maintenance** > High Availability > Backup/Recovery Settings > **Recovery Settings**.

c) In the **Media Recovery** region, select the **ARCHIVELOG Mode** check box.

d) Verify that Log Archive Filename Format contains `%t`, `%s`, and `%r`.

4. Configure redundant archive log destinations—one to the Flash Recovery Area and the other to `/u01/app/oracle/archive`.

Media Recovery

The database is currently in NOARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time is possible, but you must provide space for logs. If you change the database to ARCHIVELOG mode, you should make a backup immediately. In NOARCHIVELOG mode, you can make only cold backups and data may be lost in the event of database corruption.

☑ ARCHIVELOG Mode*

Log Archive Filename Format* `%t_%s_%r.dbf`
The naming convention for the archived log files. %s: log sequence number; %t: thread number; %S and %T: padding the filename to the left with zeroes.

Number	Archive Log Destination	Quota (512B)	Status	Type
1	/u01/app/oracle/archive/			Local
2	Archive Log Destination			Local
3				Local
4				Local
5				Local
6				Local
7				Local
8				Local
9				Local
10	USE_DB_RECOVERY_FILE_DEST	n/a	VALID	Local

☑ **TIP** It is recommended that archive log files be written to multiple locations spread across the different disks.
☑ **TIP** You can specify up to 10 archive log destinations.

Flash Recovery Area

It is highly recommended that you use flash recovery area to automate your disk backup management.

Flash Recovery Area Location `/u01/app/oracle/flash_recovery_area`

Notice that the database is preconfigured to save archived logs to the Flash Recovery Area by default (Archive Log Destination 10). Add an additional Archive Log Destination so that you will have redundant copies of your log files.

a) In **Archive Log Destination** number 1, enter `/u01/app/oracle/archive/`. The directory path should end with a slash. Leave **Quota** blank.

Note: You must create the directory, if it does not already exist. You already did this in step (3).

Solutions for Practice 14: Backup and Recovery Concepts (continued)

b) Optionally, click **Show SQL**, review the statements, and then click **Return**.

```
Show SQL


ALTER SYSTEM SET log_archive_dest_1 =
"LOCATION=/u01/app/oracle/archive/ OPTIONAL REOPEN=300" SCOPE=BOTH
ALTER SYSTEM SET log_archive_dest_10 =
"LOCATION=USE_DB_RECOVERY_FILE_DEST OPTIONAL REOPEN=300" SCOPE=BOTH
```

c) Click **Apply**. When prompted whether you want to restart the database now, click **Yes**.

d) Enter the credentials to restart the database, and then click **OK**.

e) When asked to confirm, click **Yes** again.

Now that your database is in `ARCHIVELOG` mode, it will continually archive a copy of each online redo log file before reusing it for additional redo data.

Note: Remember that this consumes space on the disk and that you must regularly back up older archive logs to some other storage.

Solutions for Practice 15: Performing Database Backups

Background: Your database is ready to move from development and test into production. Ensure that your database is configured so that recovery is possible without loss of data.

1. What is the difference between a backup set and an image copy?

    Answer: A backup set contains data and archive log files packed in an Oracle proprietary format. Files must be extracted before use. Image copies are the equivalent of operating system file copies and can be used to restore operations immediately.

2. What is the destination of any disk backups that are done?

    a) In Enterprise Manager, select **Maintenance** > High Availability > Backup/Recovery Settings > **Backup Settings**. Note the message under the Disk Backup Location that says the Flash Recovery Area is the current disk backup location.

3. Test making a backup to disk, as a backup set, with `oracle` for Host Credentials.

    a) Select **Backup Set** as your Disk Backup Type.

    b) Scroll to the bottom and enter `oracle` and `oracle` for **Host Credentials Username** and **Password** for your server.

    c) Click **Test Disk Backup**.

    d) When the test finishes, click **OK**.

Solutions for Practice 15: Performing Database Backups (continued)

4. Back up your entire database, without archived logs, while the database is open for user activity. This backup should be the base for an incremental backup strategy.

    Question: What prerequisite *must* be met to create a valid backup of a database without shutting it down?

    Answer: The database must be in ARCHIVELOG mode. Backups made with the database open but not in ARCHIVELOG mode cannot be used for recovery.

    a) Select **Maintenance** > High Availability > Backup/Recovery > **Schedule Backup**.

    b) Select **Whole Database** as the object to be backed up.

    c) Enter oracle and oracle for **Host Credentials Username** and **Password** for your server.

    d) Click **Schedule Customized Backup**.

    e) Select **Full Backup** for your Backup Type, and select the **Use as the base of an incremental backup strategy** check box.

    f) Select **Online Backup** as Backup Mode.

    g) In the **Advanced** region, deselect the **Also backup all archived logs on disk** check box, and then click **Next** to continue.

Solutions for Practice 15: Performing Database Backups (continued)

h) Select **Disk** for your backup location (notice that your Disk Backup Location is retained from step [2]). Click **Next** to continue.

i) Accept all the defaults on the Schedule page, and then click **Next** to continue.

Job

* Job Name	BACKUP_ORCL.ORACLE.COM_0|
Job Descripton	Whole Database Backup

Schedule

Time Zone GMT -7:00 ▾

Start

⦿ Immediately
○ Later

Date Jun 9, 2005
(example: Jun 9, 2005)

Time 2 ▾ 00 ▾ ⦿ AM ○ PM

Repeat

⦿ One Time Only
○ Interval

Frequency 1   Minutes ▾

○ Monthly
○ Yearly

Repeat Until

⦿ Indefinite
○ Custom

Date Jun 9, 2005
(example: Jun 9, 2005)

Time 3 ▾ 30 ▾ ⦿ AM ○ PM
(Ignored except when repeating by minutes or hours.)

j) Click **Submit Job** to perform the online database backup.

Solutions for Practice 15: Performing Database Backups (continued)

    k)   Click **View Job** to monitor the status of the backup job. This backup takes approximately five minutes to complete.

5. Schedule nightly disk-based incremental online backups for your whole database, without archived logs. Schedule it for 11:00 p.m. The schedule should be in effect indefinitely.

    a)   In Enterprise Manager, select **Maintenance** > High Availability > Backup/Recovery > **Schedule Backup**.

    b)   Select **Whole Database** as the object to be backed up.

    c)   Enter `oracle` and `oracle` for **Host Credentials Username** and **Password** for your server, and then click **Schedule Customized Backup**.

    d)   Select **Incremental Backup (Level 1)** for your Backup Type.

    e)   Select **Online Backup** as Backup Mode.

    f)   In the **Advanced** region, deselect the **Also backup all archived logs on disk** check box, and then click **Next** to continue.

    g)   Select **Disk** as your backup location, and then click **Next** to continue.

Solutions for Practice 15: Performing Database Backups (continued)

h) Change **Job Name** to `Nightly_Backup` and accept the default value for **Job Description**.

i) Select **Later** in the **Start** region. Accept today's date and use the drop-down lists and option buttons to select 11:00 p.m. for **Time**.

j) In the **Repeat** area, select **Interval**, and **Frequency 1 Days**.

k) Select **Indefinite** in the **Repeat Until** region, and then click **Next**.

l) Click **Submit Job**, and then click **OK**.

m) Navigate to **Maintenance** > Related Links > **Jobs** to see the scheduled job in the **Job Activity** list.

Solutions for Practice 16: Performing Database Recovery

Background: Many failures of the Oracle database can be traced to some sort of media failure, such as disk or controller failure. Recover your database from a variety of simulated media failures. SQL script files are provided for you in the `/home/oracle/labs` directory. If needed, use appendix C for Linux and appendix D for SQL syntax. Note that where OS file names are mentioned, your system may possibly have different file names than shown here.

1. Recover from the loss of a control file.

   a) As the `DBA1` user, run the `lab_16_01_a.sql` script to prepare some procedures to be called by the rest of this practice.

   ```
   @$HOME/labs/lab_16_01_a.sql
   ```

   b) Now run the `lab_16_01_b.sql` script. This script deletes one of your control files.

   ```
   @$HOME/labs/lab_16_01_b.sql
   ```

   c) The Help desk begins receiving calls saying that the database appears to be down. Troubleshoot and recover as necessary. Use Enterprise Manager's Database page to attempt to start up the database, and use SQL*Plus, if needed.

      - In Enterprise Manager, navigate to the Database page. It reports that the database is down and offers you the chance to start it up again. Attempt to do so by clicking **Startup**. You may need to wait one or two minutes before the page appears as described. If you see a Connection Refused message, ignore it; the connection will eventually be established.

      - Enter host credentials as `oracle` for **Username** and **Password**, and then click **OK**.

      - When asked whether you are sure that you want to start the database, click **Yes**.

Solutions for Practice 16: Performing Database Recovery (continued)

   d) The startup of the instance fails with Enterprise Manager, and you can get no other
      information to explain the failure. So use the command-line tools.

      - Connect to the instance with SQL*Plus as sysdba and check the current status of the
        instance.

```
sqlplus / as sysdba
select status from v$instance;
```

      - The instance status is STARTED, which means that the database is in the NOMOUNT
        stage. Attempt to mount the database by entering this:

```
SQL> alter database mount;
```

```
SQL> alter database mount;
alter database mount
*
ERROR at line 1:
ORA-00205: error in identifying control file, check alert log for more info
```

   e) The instance cannot move to the mount stage because it cannot find one of the control
      files. Check the last 10 rows of the alert log to see which control file is the problem.

```
SQL> host tail -10 $ORACLE_BASE/admin/orcl/bdump/aler*
```

      Output:

```
SQL> host tail -10 $ORACLE_BASE/admin/orcl/bdump/aler*
ORA-205 signalled during: ALTER DATABASE   MOUNT...
Sun Oct 30 13:35:01 2005
alter database mount
Sun Oct 30 13:35:01 2005
ORA-00202: control file:
'/u01/app/oracle/flash_recovery_area/ORCL/controlfile/o1_mf_1p4sbncs_.ct
l'
ORA-27037: unable to obtain file status
Linux Error: 2: No such file or directory
Additional information: 3
Sun Oct 30 13:35:01 2005
ORA-205 signalled during: alter database mount...
```

Solutions for Practice 16: Performing Database Recovery (continued)

f) The control file in the Flash Recovery Area is missing. Restore the missing control file by copying the existing control file, and then mount and open the database.

```
SQL> host
$ cd /u01/app/oracle/oradata/ORCL/controlfile
$ ls -l
total 6908
-rw-r-----  1 oracle  oinstall  7061504 Oct 30 13:32 o1_mf_1p4sbn1o_.ctl
$ cp o1_mf_1p4sbn1o_.ctl
/u01/app/oracle/flash_recovery_area/ORCL/controlfile/o1_mf_1p4sbncs_.ctl
$ exit
```

```
SQL> alter database mount;
```

```
SQL> alter database open;
```

g) Why did you have to use two commands to move the instance state from NOMOUNT to OPEN?

Answer: Because the ALTER DATABASE command enables you to change only one state level per command

h) Why did you use operating system commands to restore the control file instead of using Oracle Recovery Manager?

Answer: Because all control files are identical. As long as any one control file is intact, it can be used to restore the others.

Solutions for Practice 16: Performing Database Recovery (continued)

2. Recover from the loss of an application data file.

   a) Start a SQL*Plus session as the DBA1 user, and run the lab_16_02.sql script. This script deletes one of your application data files.

   ```
   $ sqlplus dba1/oracle @$HOME/labs/lab_16_02.sql
   ```

   b) The Help desk has received a call from a user who is unable to access the COUNTRIES table in the HR application schema. Count the rows in the table to see whether there is a problem.

   ```
   SQL> select count(*) from HR.COUNTRIES;
   select count(*) from HR.COUNTRIES
                               *
   ERROR at line 1:
   ORA-01116: error in opening database file 5
   ORA-01110: data file 5:
   '/u01/app/oracle/oradata/ORCL/datafile/o1_mf_example_1p4sd3y2_.dbf'
   ORA-27041: unable to open file
   Linux Error: 2: No such file or directory
   Additional information: 3
   ```

   c) Troubleshoot and recover as necessary. The error message suggests that the data file for the EXAMPLES tablespace is corrupt or missing. Using operating system commands, verify that there is a problem with the file.

   ```
   SQL> host ls
   /u01/app/oracle/oradata/ORCL/datafile/o1_mf_example_1p4sd3y2_.dbf
   ls: /u01/app/oracle/oradata/ORCL/datafile/o1_mf_example_1p4sd3y2_.dbf:
   No such file or directory
   ```

   d) Recover the data file to the current time, specifying the missing data file to be recovered.

      - In Enterprise Manager, select **Maintenance** > High Availability > Backup/Recovery > **Perform Recovery**.

      - Select **Datafiles** from the **Object Type** drop-down list.

      - In the Object Level Recovery region, select **Recover to current time** for the **Operation Type**.

      - Enter the host credentials as oracle and oracle for **Username** and **Password**, and then click **Perform Object Level Recovery**.

Solutions for Practice 16: Performing Database Recovery (continued)

- On the Datafiles page, select the data file in question.

Select	Datafile Name	Datafile Number	Status	Enabled
☑	/u01/app/oracle/oradata/ORCL/datafile/o1_mf_example_1p4sd3y2_.dbf	5	ONLINE	READ WRITE

- Click **Next**.

- Because the problem is simply a deleted file rather than a bad hard drive, there is no need to restore the file to a different location. Select **No. Restore the files to the default location**, and then click **Next**.

- Click **Submit**. (*It will take one or two minutes for the operation to complete.*)

e) When you see the Operation Succeeded message, ensure that the restored data file is online.

- In Enterprise Manager, select **Administration** > Database Administration > Storage > **Datafiles**.

Datafiles

Object Type Datafile

Search
Select an object type and optionally enter an object name to filter the data that is displayed in your results set.
Object Name [   ]
(Go)
By default Datafiles are case-sensitive searches. To run an exact match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

(Create)

(Edit) (View) (Delete) Actions Create Like ▾ (Go)

Select	File Name △	Tablespace	Status	Size (MB)	Used (MB)	Used (%)
⦿	/u01/app/oracle/oradata/ORCL/datafile/inventory01.dbf	INVENTORY	ONLINE	50.000	42.188	84.38
○	/u01/app/oracle/oradata/ORCL/datafile/o1_mf_example_1pbkt0j1_.dbf	EXAMPLE	ONLINE	100.000	68.250	68.25
○	/u01/app/oracle/oradata/ORCL/datafile/o1_mf_sysaux_1p4s741n_.dbf	SYSAUX	ONLINE	270.000	264.812	98.08
○	/u01/app/oracle/oradata/ORCL/datafile/o1_mf_system_1p4s740z_.dbf	SYSTEM	SYSTEM	490.000	480.688	

f) Verify that the COUNTRIES table is now accessible.

```
SQL> select count(*) from HR.COUNTRIES;

  COUNT(*)
----------
        25
```

Solutions for Practice 16: Performing Database Recovery (continued)

3. Recover from the loss of a system data file.

   a) Why is recovery from the loss of a system data file or a data file belonging to an undo tablespace different from recovering an application data file?

   Answer: Because recovery of system or undo data files must be done with the database closed, whereas recovery of an application data file can be done with the database open and available to users.

   b) As SYSDBA, run the `lab_16_03.sql` script. This script deletes the system data file.

```
SQL> @lab_16_03.sql

PL/SQL procedure successfully completed.

ORACLE instance shut down.
SQL>
```

   c) In Enterprise Manager, review the Database home page. The database is shut down, so you click **Startup** to try to open it. If you see a message that says the connection was refused, dismiss it, and reenter the EM home page URL in the browser.

   - Enter the host credentials as `oracle` and `oracle` for the host **Username** and **Password**, and enter `DBA1` and `oracle` for the database credentials, and then click **OK**.

   - Click **Yes**.

Solutions for Practice 16: Performing Database Recovery (continued)

    d) This command will fail with the database left in the MOUNT state, because there is a data file missing from the SYSTEM tablespace.

    e) Click **Perform Recovery**.

- If prompted, enter host (oracle/oracle) and database (dba1/oracle AS SYSDBA) credentials, and then click **Continue**.

- In the **Object Level Recovery** region, select **Datafiles** for Object Type.

- Select **Recover to current time** for **Operation Type**.

- Fill in the host credentials if not already set, and then click **Perform Object Level Recovery**.

- Select the data file for the SYSTEM tablespace, and then click **Next**.

- Because the problem is simply a deleted file rather than a bad hard drive, there is no need to restore to a different location. Select **No. Restore the files to the default location**, and then click **Next**.

- Click **Submit**. It will take three to four minutes for the operation to complete.

    f) When you see the Operation Succeeded message, click **Open Database**.

    g) After you see the success message, click **OK**, and then verify that the database is open and operating normally by logging into EM as DBA1/oracle, as SYSDBA, and reviewing the Database home page.

Solutions for Practice 17: Performing Flashback

Background: You decide to gain hands-on experience in some of the flashback functionality. To avoid impacting other users, you will first copy the DEPARTMENTS table of the HR schema to DEPARTMENTS2.

1. Log in to SQL*Plus as DBA1 user and create a new HR.DEPARTMENTS2 table based on the HR.DEPARTMENTS table.

    a) Log in to SQL*Plus as dba1/oracle.

    ```
    $ sqlplus dba1/oracle
    ```

    b) Enter the following command to create the copy table:

    ```
    SQL> create table hr.departments2 as select * from hr.departments;
    ```

    c) Count the rows in the DEPARTMENTS2 table. There should be 27 rows.

    ```
    SQL> SELECT COUNT(*) FROM HR.DEPARTMENTS2;

      COUNT(*)
    ----------
            27
    ```

2. Drop the HR.DEPARTMENTS2 table, and then verify that it has indeed been dropped.

    ```
    SQL> DROP TABLE HR.DEPARTMENTS2;

    Table dropped.

    SQL> SELECT * FROM HR.DEPARTMENTS2;
    SELECT * FROM HR.DEPARTMENTS2
                    *
    ERROR at line 1:
    ORA-00942: table or view does not exist
    ```

3. Use the FLASHBACK TABLE command to restore the table. Count the rows in the DEPARTMENTS2 table.

    ```
    SQL> FLASHBACK TABLE hr.departments2 TO BEFORE DROP;

    Flashback complete.

    SQL> SELECT COUNT(*) FROM hr.departments2;

      COUNT(*)
    ----------
            27
    ```

Solutions for Practice 17: Performing Flashback (continued)

4. Run the `lab_17_04.sql` script to insert three rows into the `HR.DEPARTMENTS2` table by using three separate transactions. The new rows have `DEPARTMENT_ID` values of `280`, `290`, and `300`.

```
SQL> @lab_17_04.sql
```

5. Use Enterprise Manager to perform flashback to the version of the table where only the first of the three new rows is present (with `DEPARTMENT_ID = 280`). First, evaluate row changes to decide on a point in time. Limit your analysis to the new rows just added: where `department_id >= 280`. If you receive an error while performing the flashback, you may need to enable row movement on the table. See the next step.

a) In Enterprise Manager, select **Maintenance** > High Availability > Backup/Recovery > **Perform Recovery**.

b) Select Tables from the **Object Type** drop-down list, and then select **Flashback Existing Tables** for **Operation Type**. Click **Perform Object Level Recovery**.

c) Select **Evaluate row changes and transactions to decide on a point in time**, and enter `HR.DEPARTMENTS2` as the fully qualified name of the table in the **Table** field, and then click **Next**.

Solutions for Practice 17: Performing Flashback (continued)

    d) Highlight `DEPARTMENT_ID` under **Available Columns,** and then click the **Move** button to move it under **Selected Columns**. Under Step 2, enter a `WHERE` clause that will select the added rows. For example, earlier you added rows with `DEPARTMENT_IDs` of `280`, `290`, and `300`, so the "`WHERE department_id >= 280`" clause is suitable.

    e) Click **Next** to continue.

    f) You can now review the rows under Flashback Versions Query Result.

Flashback Versions Query Result

Select	Flashback SCN	Flashback Timestamp	Transaction ID	Operation	DEPARTMENT_ID
●	6800080	Jun 9, 2005 5:15:32 PM	0A002B00A6030000	INSERT	300
○	6800061	Jun 9, 2005 5:15:15 PM	03002400CB030000	INSERT	290
○	6800049	Jun 9, 2005 5:14:53 PM	05000600FA030000	INSERT	280

    g) Under **Flashback Versions Query Result**, select the middle transaction from the list to flash back. Click **Next** to continue.

Solutions for Practice 17: Performing Flashback (continued)

h) On the next page, you are asked to include any tables related to or dependent on the table that you are flashing back. Because there are none, click **Next** to continue.

i) Review the SQL statement that you are about to execute. Click **Show SQL**, view the SQL statement, click **OK**, and then click **Submit**.

```
FLASHBACK TABLE HR.DEPARTMENTS2 TO SCN 6800061
```

6. On the next page, you find that the operation has failed because row movement is not enabled for the table. You may recall from the lesson that row movement must be enabled for this feature to work.
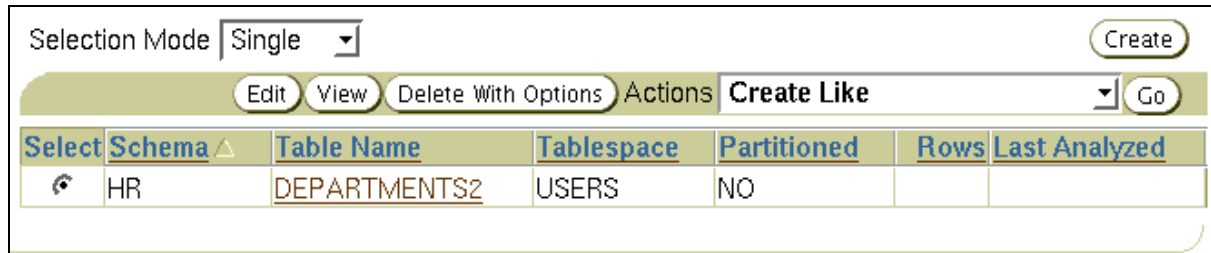
a) Note this error:

b) To enable row movement for this table, select **Administration** > Schema > Database Objects > **Tables**.

c) Enter HR in the **Schema** field and DEPARTMENTS2 in the **Object Name** field, and then click **Go**.

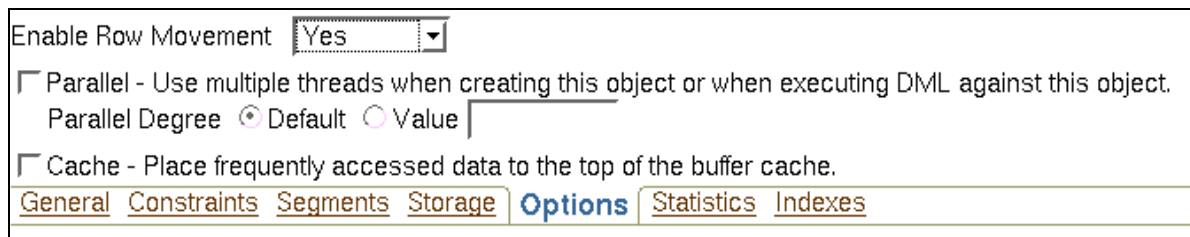## Solutions for Practice 17: Performing Flashback (continued)

d) One table is displayed in the **Results** region and is selected by default. Click **Edit**.

Select	Schema △	Table Name	Tablespace	Partitioned	Rows	Last Analyzed
⊙	HR	DEPARTMENTS2	USERS	NO		

Selection Mode Single ▾   Create

Edit View Delete With Options Actions Create Like ▾ Go

The Edit Table page appears.

e) Click the **Options** tab.

f) From the **Enable Row Movement** drop-down list, select **Yes**.

Enable Row Movement Yes ▾
☐ Parallel - Use multiple threads when creating this object or when executing DML against this object.
   Parallel Degree ⊙ Default ○ Value
☐ Cache - Place frequently accessed data to the top of the buffer cache.

General Constraints Segments Storage | **Options** | Statistics Indexes

g) Click **Show SQL**, and then click **Return**.

```
ALTER TABLE "HR"."DEPARTMENTS2" ENABLE ROW MOVEMENT
```

h) Click **Apply** to apply the change.

After the table modification is confirmed, you can perform the flashback operation.

Repeat step 5 to perform the version flashback of the table. Because row movement has been enabled, it should succeed this time.

Confirmation

The selected tables, HR.DEPARTMENTS2, have been flashed back.

Solutions for Practice 17: Performing Flashback (continued)

i) In SQL*Plus, count the rows of the HR.DEPARTMENTS2 table to confirm the
    flashback operation. Note that there is only one additional row now, not three. Then
    display the rows where DEPARTMENT_ID >= 280. Note that only one of the original
    three is remaining.

```
SQL> select count(*) from hr.departments2;

  COUNT(*)
----------
        28

SQL> select * from hr.departments2 where department_id >= 280;

DEPARTMENT_ID DEPARTMENT_NAME                        MANAGER_ID LOCATION_ID
------------- ----------------------------- ---------- -----------
          280 DUMMY1

SQL>
```

Solutions for Practice 18: Moving Data

Background: In the recent past, you received a number of questions about the HR schema. To analyze them, without interfering in the daily activities, you decide to use the Data Pump Wizard to export the HR schema to file. When you perform the export, you are not sure into which database you will be importing this schema.

In the end, you find out that the only database for which the management approves an import for, is the ORCL database. So you perform the import with the Data Pump Wizard, remapping the HR schema to a newly created HR_TEST schema in the HR_TEST tablespace. To follow best practice guidelines, you also decide to create a DP user who will be a DBA performing Data Pump jobs. For your convenience in class, the creation of the DP user is included in the lab_18_01.sql script. SQL scripts are in the /home/oracle/labs directory. However, there is one step missing.

Then, you receive two data load requests for which you decide to use SQL*Loader.

1. Review the lab_18_01.sql script, which creates the HR_TEST tablespace, the HR_TEST_ROLE role, and the HR_TEST and DP users.

   a) Double-click the **oracle's Home** icon on your desktop, double-click **labs**, and then double-click **lab_18_01.sql** to review this script.

   b) Note the passwords for these users.

      HR_TEST password: **hr_4test**

      DP password: **dp_4test**

   c) Which additional step do you need to perform to allow the DP user access to Enterprise Manager as Administrator?

   You need to log in to Enterprise Manager as the SYS user in the SYDBA role and make the DP user an EM Administrator (as you learned in the lesson titled "Managing the Oracle Instance").

   d) Close the lab_18_01.sql window.

2. Execute the lab_18_02.sh script. Then, perform the required step to make the DP user an EM administrator.

   a) Right-click your desktop and select **Open Terminal**. Then navigate to the /home/oracle/labs directory by entering **cd labs**.

   b) Enter **./lab_18_02.sh** to create the HR_TEST tablespace, the HR_TEST_ROLE role, and the HR_TEST and DP users.

   c) Log in to Enterprise Manager as the SYS user in the SYSDBA role.

Solutions for Practice 18: Moving Data (continued)

    d) Make the `DP` user an EM Administrator (as you learned in the lesson titled "Managing the Oracle Instance"). (Select Setup > Create > Enter Name: `DP`, Password: `dp_4test`, Super Administrator, checked > Click Finish > Finish > Logout.)

3. Log in to Enterprise Manager as the `DP` user in the `Normal` role and export the `HR` schema.

    a) Invoke Enterprise Manager as the `DP` user in the `Normal` role for your `ORCL` database. The **Connect As** setting should be Normal.

If this is the first time that the `DP` user logs in, click **I agree** to accept the licensing agreement.

    b) Select **Maintenance** > Data Movement > Move Row Data > **Export to Export Files**.

    c) Select **Schemas**, enter **oracle** as **Username** and **Password**, select **Save as Preferred Credential**, and then click **Continue**.

    d) On the Export: Schemas page, click **Add**, select the `HR` schema, and then click the **Select** button.

    e) Click **Next**.

Solutions for Practice 18: Moving Data (continued)

f) On the Export: Options page, select DATA_PUMP_DIR from the **Directory Object** drop-down list, and enter **hrexp.log** as **Log File**.

g) Review Advanced Options, but do not change, and then click **Next**.

h) On the Export: Files page, select DATA_PUMP_DIR from the **Directory Object** drop-down list, enter **HREXP%U.DMP** as **File Name**, and then click **Next**.

i) On the Export: Schedule page, enter **hrexp** as **Job Name** and **Export HR schema** as **Description**, accept the immediate job start time, and then click **Next**.

j) On the Export: Review page, click **Show PL/SQL** and review the PL/SQL that the Export Wizard helped you to create.

k) Click **Submit Job**.

A processing message appears, and then a success message. If not, resolve any errors, which may have occurred.

l) When the Job Activity Confirmation page appears, click the HREXP job name, and then monitor the job progress by clicking the browser's **Reload** button.

Note: Please wait, not only for the job to be created, but also for the job to complete execution. (It may take two minutes.)

4. As the DP user, import the exported HR schema back into the ORCL database, remapping it to the previously created HR_TEST schema.

Solutions for Practice 18: Moving Data (continued)

a) Invoke Enterprise Manager as the DP user in the Normal role for your ORCL database.

b) Select **Maintenance** > Data Movement > Move Row Data > **Import from Export Files**.

c) On the Import: Files page, select DATA_PUMP_DIR from the **Directory Object** drop-down list, and enter **HREXP%U.DMP** as **File Name**.

d) Select **Schemas** as the Import Type.

e) Confirm oracle as your **Host Credentials**, and then click **Continue**.

At this point, the export file is read, to verify the contents. Wait for this to complete.

f) On the Import: Schemas page, click **Add**, select HR, and then click **Select**.

g) Click **Next**.

h) On the Import: Re-Mapping page, click **Add Another Row** under **Re-Map Schemas**. Then select **HR_TEST** as **Destination Schema**.

i) Click **Add Another Row** under **Re-Map Tablespaces**, enter **HR_TEST** as **Destination Tablespace.**

j) Click **Next**.

Solutions for Practice 18: Moving Data (continued)

k) On the Import: Options page, select **DATA_PUMP_DIR** from the **Directory Object** drop-down list, enter **hrimport.log** as **Log File**, review the advanced options, but leave them at their default values, and then click **Next**.

l) On the Import: Schedule page, enter **hrimp** as **Job Name** and **Import HR schema for test purposes** as **Description**.

m) Select **Later** as **Start** and enter a time between 2 and 5 minutes from now (to give yourself time for the following steps):

- Click **Next**.

- On the Import: Review page, review the PL/SQL that the Data Pump Wizard creates for you, and then click **Submit Job**.

- After the confirmation that the job was successfully created, note the job name (for example HRIMP), and navigate to **Maintenance** > Data Movement > Move Row Data > **Monitor Export and Import Jobs**.

- Click your last job (for example, HRIMP).

- On the "Monitor Data Pump job" page, click **Reload** in your browser, when it is time to execute your job.

Monitor Data Pump Job				
		Page Refreshed **Jun 10, 2005 5:25:06 PM** (OK)		
Job Name	IMPORT000024			
Job Status	**EXECUTING** (Change Job State)			
Percent Done	**0**			
	Percent done is for table data only.			
Non-fatal Error Count	**0**			
Log File				
Maximum Number of Threads in Export Job	**1** (Change Parallelism)			

Objects Currently Being Imported

The number of rows in this table corresponds to the degree of parallelism.

Name	Type	Percent Done
		0

Import Files

File Name	Maximum File Size (MB)	Used (MB)	Used (%)
/u01/app/oracle/product/10.2.0/db_1/rdbms/log/HREXP01.DMP	Auto Extend	0	0.0

Solutions for Practice 18: Moving Data (continued)

> *- Optional,as this requires quick action:* Click **Change Job State**.

> *- Optional (continued):* Click **Suspend** and later **Resume** to halt the job and then to continue it again.

n) In the end, you want to see that your job executed 100% without any errors.

o) Click **OK**.

p) Verify that the import succeeded by viewing the log file.

```
$ cat </u01/app/oracle/product/10.2.0/db_1/rdbms/log/hrimport.log
```

Note: You may see an error saying that the hr_test object already exists. This is because that user existed when you did the export, and still exists. This is not a problem.

q) Using SQL*Plus, connect to the database as the HR_TEST user.

```
$ sqlplus hr_test/hr_4test
```

r) Select data from tables in the hr_test schema, for verification of the import.

```
SQL> select * from jobs;
```

Solutions for Practice 18: Moving Data (continued)

5. As the `DP` user, load data into the `PRODUCT_MASTER` table by using SQL*Loader via Enterprise Manager Database Control.

    a) Invoke Enterprise Manager as the `DP` user in the `Normal` role for your `ORCL` database.

    b) Select **Maintenance** > Data Movement > Move Row Data > **Load Data from User Files**.

    c) Click **Use Existing Control File**.

    d) If you have not done so before, enter **oracle** as **Username** and as **Password**, click **Save as Preferred Credential**, and then click **Continue**.

    e) On the Load Data: Control File page, enter **/home/oracle/labs/lab_18_05.ctl** as control file name and path, or use the flashlight icon to select this control file. Click **Next**.

    f) On the Load Data: Data File page, click **Provide the full path and name on the database server machine** and enter **/home/oracle/labs/lab_18_05.dat** as data file name and path, or use the flashlight icon to select this data file. Click **Next**.

    g) On the Load Data: Load Method page, accept **Conventional Path**, and then click **Next**.

    h) On the Load Data: Options page, accept all defaults, but enter **/home/oracle/labs/lab_18_05.log** as log file name and path.

    i) Review the advanced options, but do not change any, and then click **Next**.

Solutions for Practice 18: Moving Data (continued)

j) On the Load Data: Schedule page, enter **lab_18_05** as **Job Name** and **Load data into the PRODUCT_MASTER table** as **Description**. Let the job start immediately, and then click **Next**.

k) On the Load Data: Review page, review the loading information and parameters, and then click **Submit Job**.

l) Confirm your results by viewing your lab_18_05.log file in your /home/oracle/labs directory.

6. As the INVENTORY user, load data into the PRODUCT_ON_HAND table by using SQL*Loader command line.

a) Invoke a terminal window and navigate to the /home/oracle/labs directory.

b) Enter the following SQL*Loader command (in continuation, without pressing [Enter] before reaching the end of the command): **sqlldr userid=inventory/verysecure control=lab_18_06.ctl log=lab-_18_06.log data=lab_18_06.dat**.

```
sqlldr userid=inventory/verysecure control=lab_18_06.ctl
log=lab_18_06.log data=lab_18_06.dat

SQL*Loader: Release 10.2.0.1.0 - Production on Tue Aug 02 22:24:44
2005

Copyright © 1982, 2005, Oracle. All rights reserved.

Commit point reached - logical record count 64
Commit point reached - logical record count 82
Commit point reached - logical record count 83
```

Solutions for Practice 18: Moving Data (continued)

    c) Confirm your results by viewing your `lab_18_06.log` file in your
        `/home/oracle/labs` directory.

Congratulations!

You completed all practices for the

Oracle Database 10g: Administration Workshop I