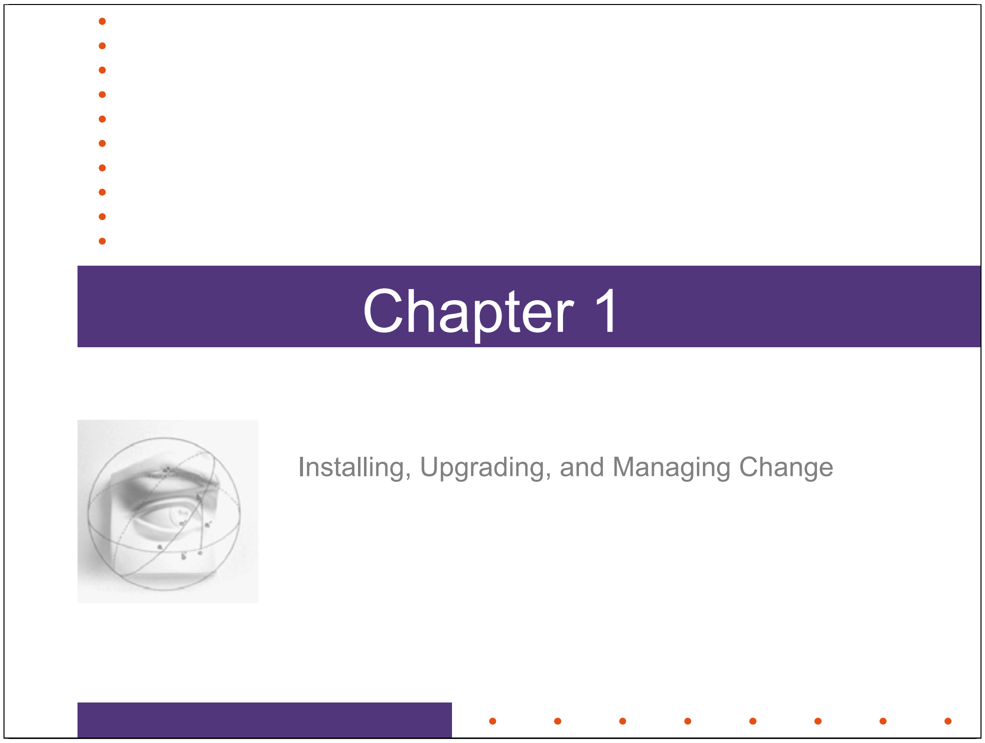
Oracle 11*g* New Features

by

Required text for this course:

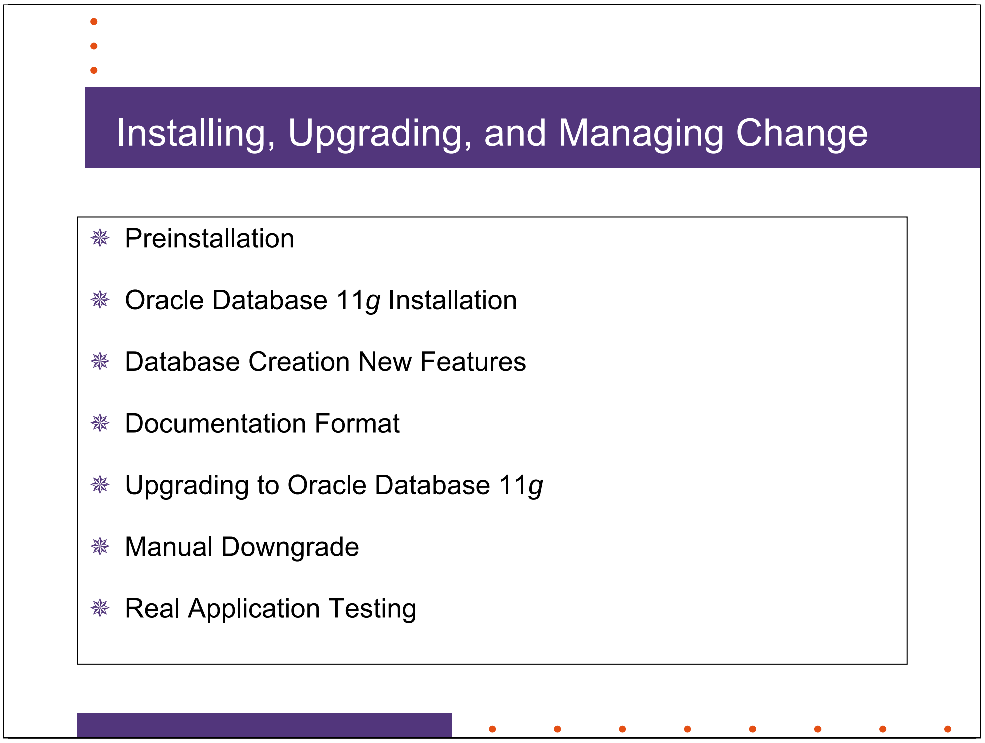
Alapati, Sam and Kim, Charles. 2007. ***Oracle Database 11g New Features for DBAs and Developers***, Apress, Berkeley, CA. ISBN 1590599101.

The instructor and each student will have a hardcopy.



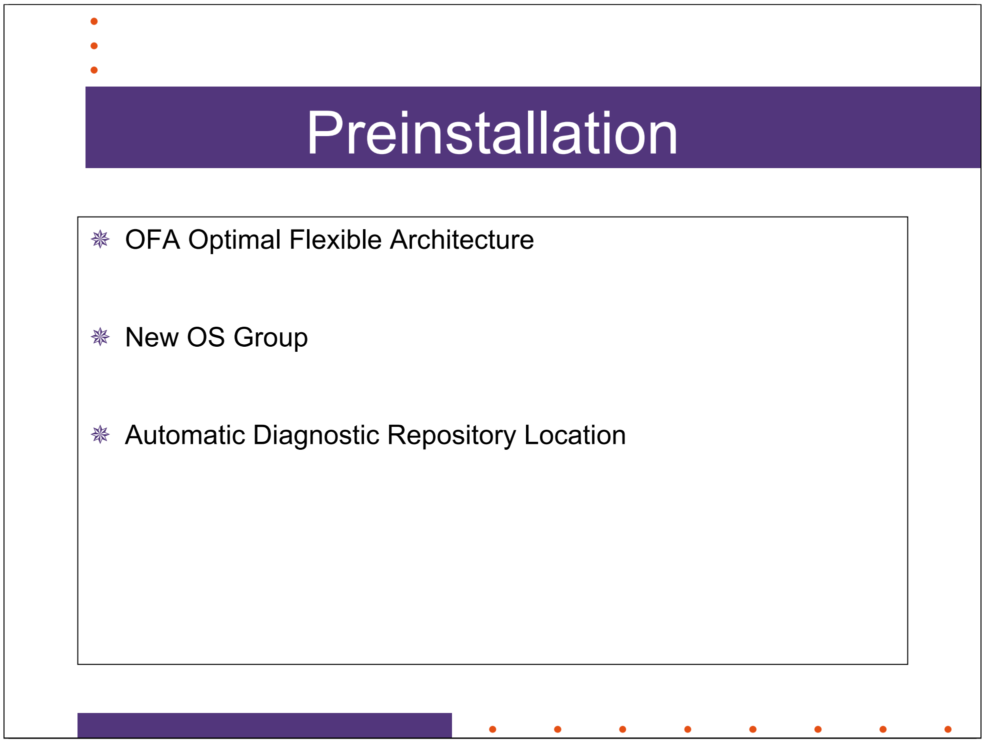
Chapter 1

Installing, Upgrading, and Managing Change



Installing, Upgrading, and Managing Change

- Preinstallation
- Oracle Database 11*g* Installation
- Database Creation New Features
- Documentation Format
- Upgrading to Oracle Database 11*g*
- Manual Downgrade
- Real Application Testing



Preinstallation

- OFA Optimal Flexible Architecture
- New OS Group
- Automatic Diagnostic Repository Location

Text Page 2-3, 5

OFA Optimal Flexible Architecture

Set the *ORACLE_BASE* environment variable (will be mandatory in future releases).

There will be a warning in ALERT LOG when Oracle Base is not set.

Don’t specify an Oracle Clusterware home under the *ORACLE_BASE* variable as it returns an error.

Oracle Clusterware and *ORACLE_BASE* need to be at the same level.

Flash Recovery Area and *oradata* directory are just under the *ORACLE_BASE* path.

New SYSASM privilege provides separate admin privilege for ASM instance.

Separates DBA role from Storage Administrator Role.

Can be conveyed to OS user by membership in “OSASM” OS group.

On UNIX/Linux, the standard name for the OSASM group is *asmadmin*.

Can be administered through *orapwd* file.

New OFA layout consolidates admin file dump destinations for all instances and Oracle products.

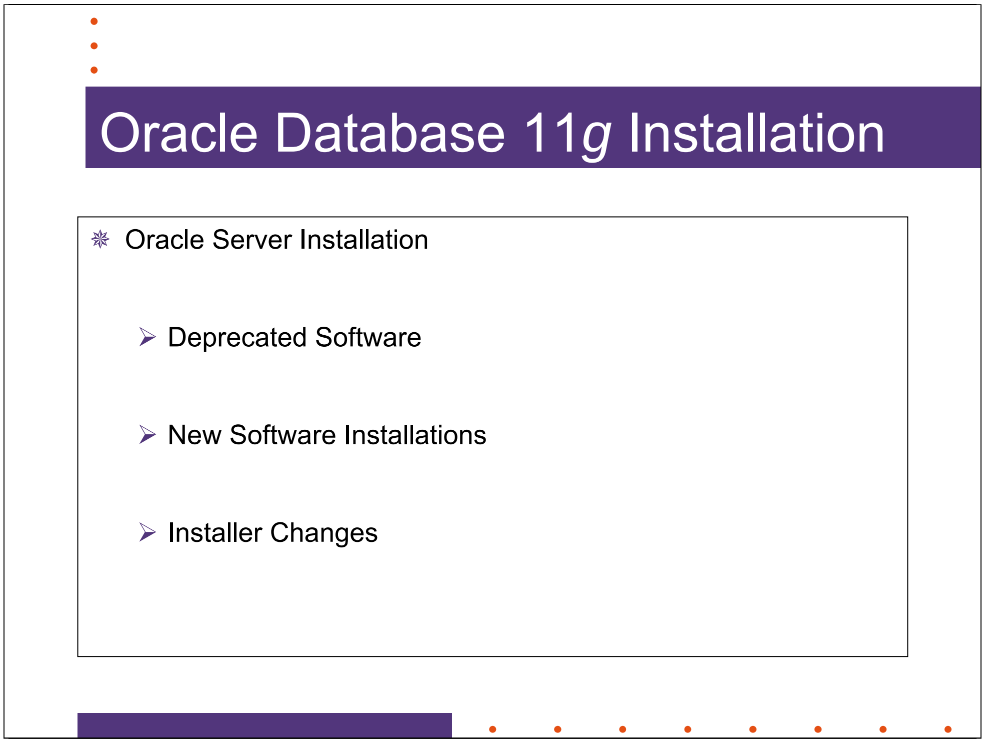
Default location under *ORACLE_BASE*.

Main directory name is *diag*.

Subdirectory for each Oracle product – databases under *rdbms*.

Database dumps under *dbname/instance_name* */u01/app/oracle/diag/rdbms/somedb/somedb/*.

Subdirectories are *cdump*, *trace*, *alert*.



Oracle Database 11*g* Installation

- Oracle Server Installation
  - Deprecated Software
  - New Software Installations
  - Installer Changes

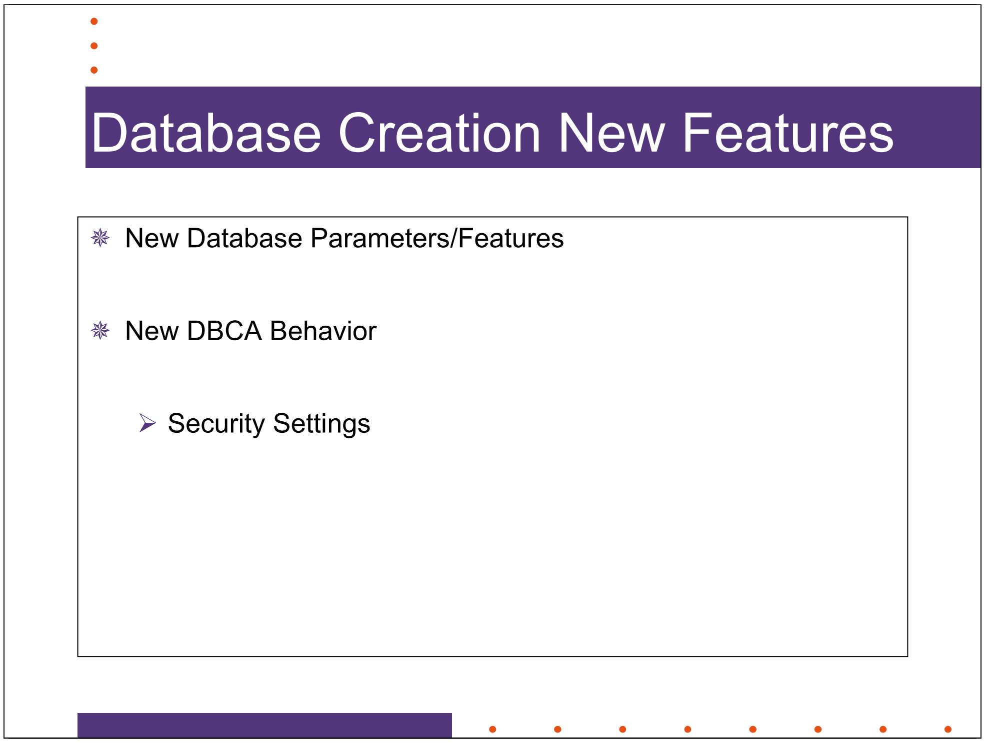
Text Page 4

Deprecated Software
- iSQL*Plus
- Oracle Enterprise Manager Java Console

New Software Installations
- APEX
- SQL Developer
- Oracle Real Application Testing
- Database Replay Tool
- SQL Performance Analyzer
- Oracle Database Vault
  - Provides a baseline of security settings.
  - No longer on Companion CD.
  - Optional Product.

Installer Changes
- Installer warns if Flash Recovery and *oradata* are in the same mount point.
- Installer specifically has a labeled area to specify the *ORACLE_BASE* and *ORACLE_HOME*.
  - Not just called **PATH**.
- Privileged Operating System Groups
  - Select new OSASM group, if you created one.
    - Defaults to *oinstall.*

Configuration Manager
- If you select "Create a Database" during software installation, installer presents an Oracle Configuration Manager Registration page.
- Can set up later with *$ORACLE_HOME/ccr/bin/setupCCR*.
- Captures information about software installed on the system.
- See ***Configuration Manager Installation and Administration Guide*** in online docs.



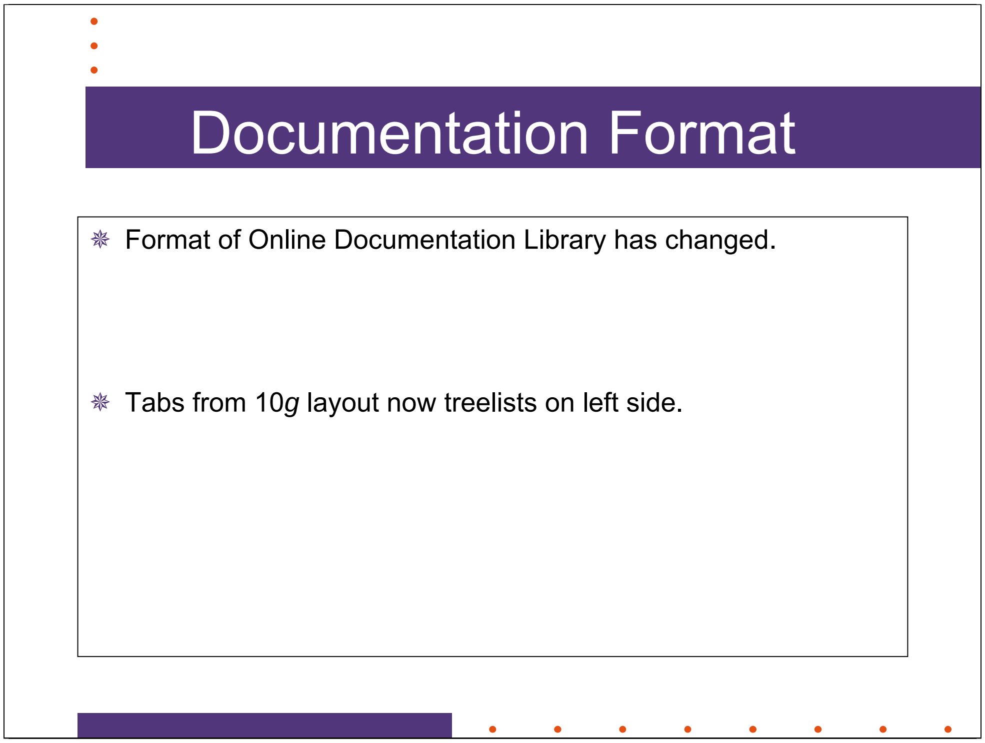
Database Creation New Features

- New Database Parameters/Features
- New DBCA Behavior
  - Security Settings

Text Page 8-17
- Text Page 8 – Memory Related Parameters
- Text Page 9 – PL/SQL Native Compilation Parameter
- Text Page 10 – The **diagnostic_dest** Parameter
- Text Page 11 – New Result Cache-Related Parameters
- Text Page 12 – Parameter to Control DDL Lock Timeout
  - SecureFiles Related Parameter – **db_ultra_safe**
- Text Page 12 – Optimizer Related Parameters
- Text Page 13 – DBCA Enhancements
- Text Page 13 – Security-Related Parameters
  - **sec_case_sensitive_logon**
- Text Page 16 – Background Processes
- Text Page 17 – New PL/SQL Packages

Miscellaneous Parameters/Features
- Create **PFILE/SPFILE** from memory.
- **Memory_Target** and **Memory_Max_Target** parameters
  - Replace **SGA_TARGET** and **PGA_AGGREGATE_TARGET**.
- Native Compilation without C compiler.
- Results Cache Query results cached in memory.
- DDL Lock Timeout
- One parameter for enabling extra database consistency checking.
  - **db_ultra_safe**
  - **db_block_checking**
  - **db_block_checksum**
  - **db_lost_write_protect**
    - Protects from O/S writes that do not actually get written to disk.

DBCA
- Security Settings



Documentation Format

- Format of Online Documentation Library has changed.
- Tabs from 10*g* layout now treelists on left side.

Students should now locate the Oracle Database Online Documentation Library 11*g*.

http://tahiti.oracle.com/
http://www.oracle.com/technology/documentation

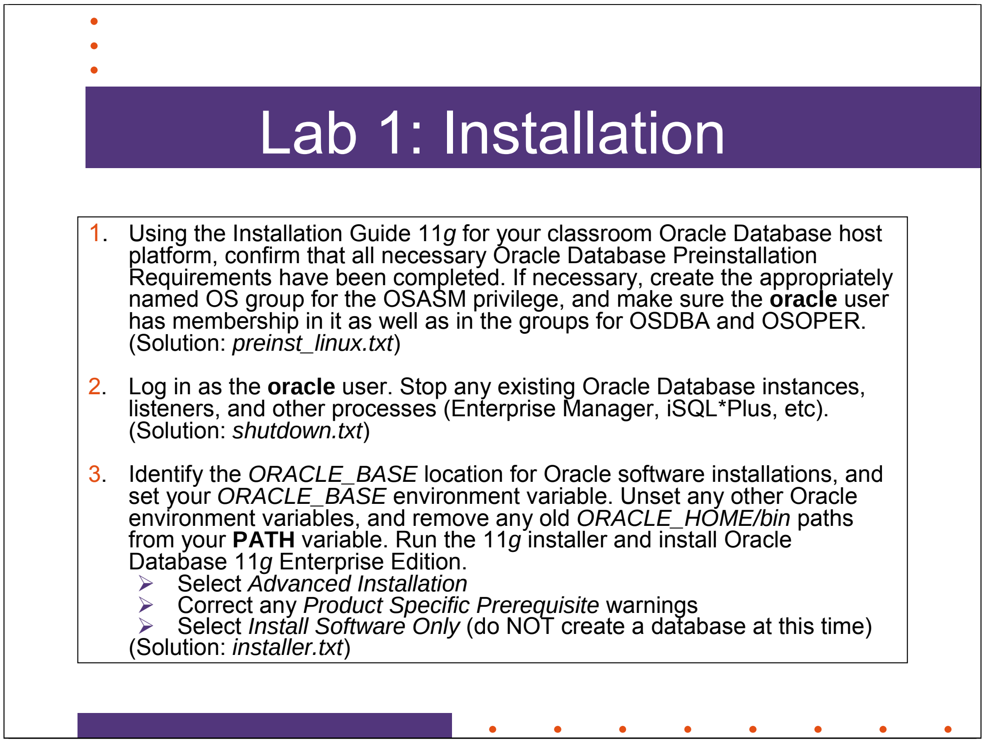
Tabs from 10*g* layout now appear as treelists in the left-side frame.

Before continuing to the next topic in class, students should bookmark the main page. Also find and bookmark:

New Features Guide
Installation Guide for Linux (or your classroom platform, if different)
Upgrade Guide

Other useful books for this class:

SQL Reference (syntax updates)
Reference (parameters and Data Dictionary)



Lab 1: Installation

1. Using the Installation Guide 11*g* for your classroom Oracle Database host platform, confirm that all necessary Oracle Database Preinstallation Requirements have been completed. If necessary, create the appropriately named OS group for the OSASM privilege, and make sure the **oracle** user has membership in it as well as in the groups for OSDBA and OSOPER. (Solution: *preinst_linux.txt*)
2. Log in as the **oracle** user. Stop any existing Oracle Database instances, listeners, and other processes (Enterprise Manager, iSQL*Plus, etc). (Solution: *shutdown.txt*)
3. Identify the *ORACLE_BASE* location for Oracle software installations, and set your *ORACLE_BASE* environment variable. Unset any other Oracle environment variables, and remove any old *ORACLE_HOME/bin* paths from your **PATH** variable. Run the 11*g* installer and install Oracle Database 11*g* Enterprise Edition.
   - ➢ Select *Advanced Installation*
   - ➢ Correct any *Product Specific Prerequisite* warnings
   - ➢ Select *Install Software Only* (do NOT create a database at this time)

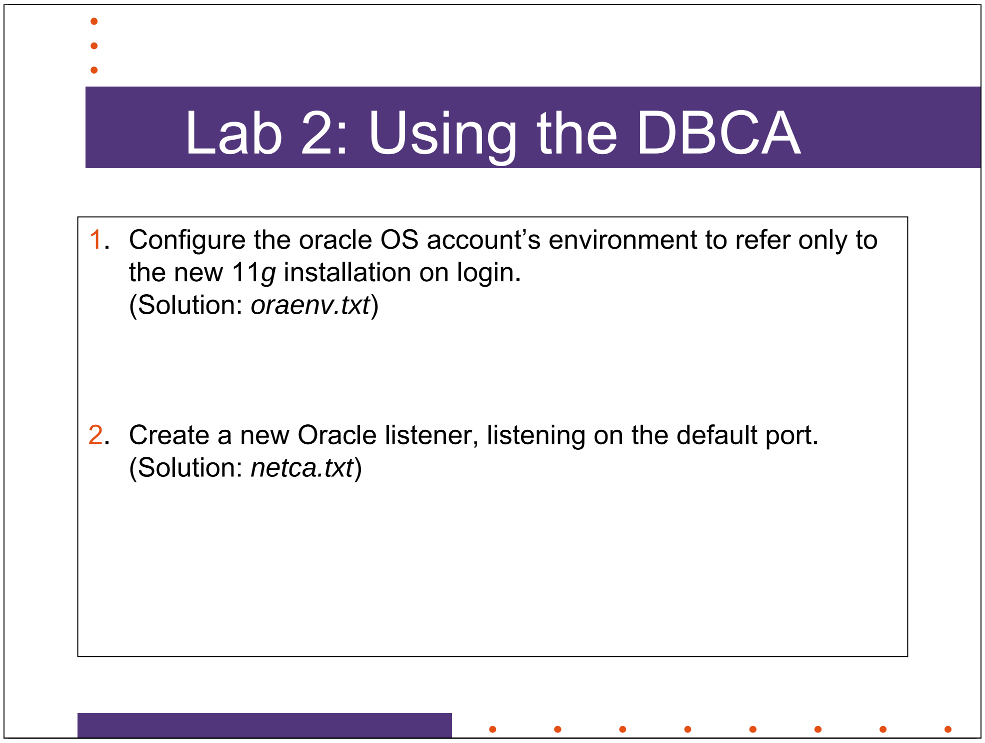
   (Solution: *installer.txt*)

Installation of 11*g*:

If the installer's “Select Installation Method” or “Install Location” page displays the old 10*g* database home, make sure the student exits the installer, unsets the *ORACLE_HOME* and *ORACLE_SID* environment variables, and removes the old 10*g* *ORACLE_HOME/bin* from the **PATH** variable before continuing.

At the “Select Configuration Option” page, make sure each student selects “Install Software Only.” We will run the DBCA to create an 11*g* database in a later lab.

Students will need the **root** password in order to run the *root.sh* script. When they do so, they should answer “Y” to each prompt.



Lab 2: Using the DBCA

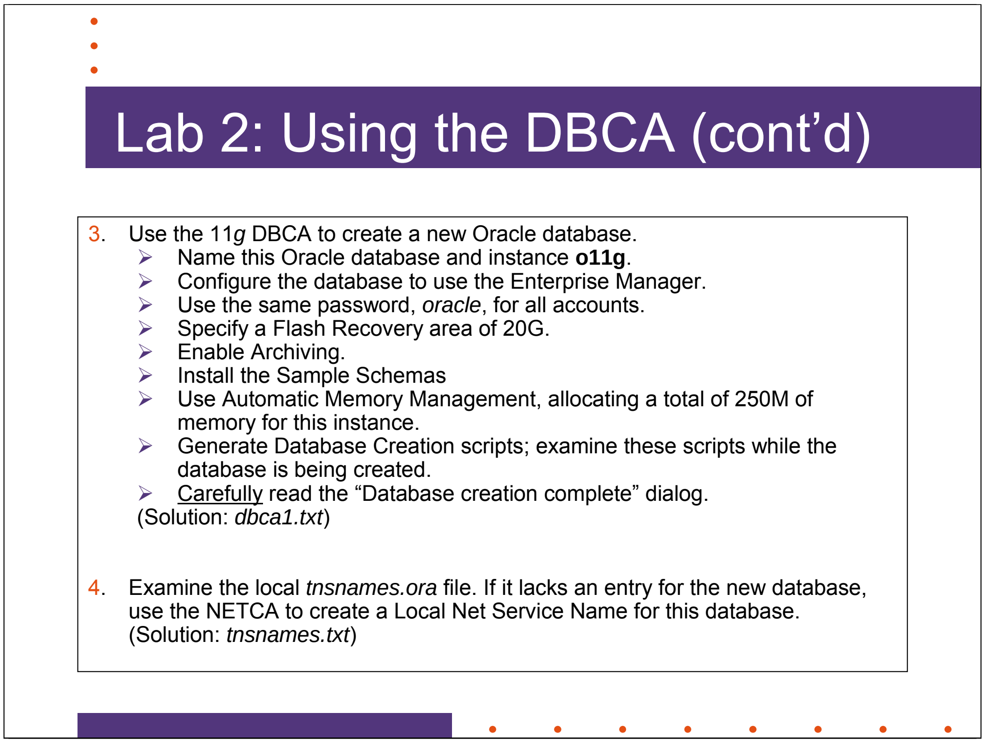
1. Configure the oracle OS account’s environment to refer only to the new 11*g* installation on login.
   (Solution: *oraenv.txt*)

2. Create a new Oracle listener, listening on the default port.
   (Solution: *netca.txt*)

Note:

All processes, including listeners, instance background processes, EM servers, iSQL*Plus servers, etc. from a previous (10*g*) Oracle installation should have already been stopped before beginning the 11*g* installation.

Lab 2 ─ Configuring a new database to use Enterprise Manager requires a running listener. Complete Exercise 2 before beginning Exercise 3. With no older listener running, this exercise is a quick “just click [Next]” trip through the NETCA.



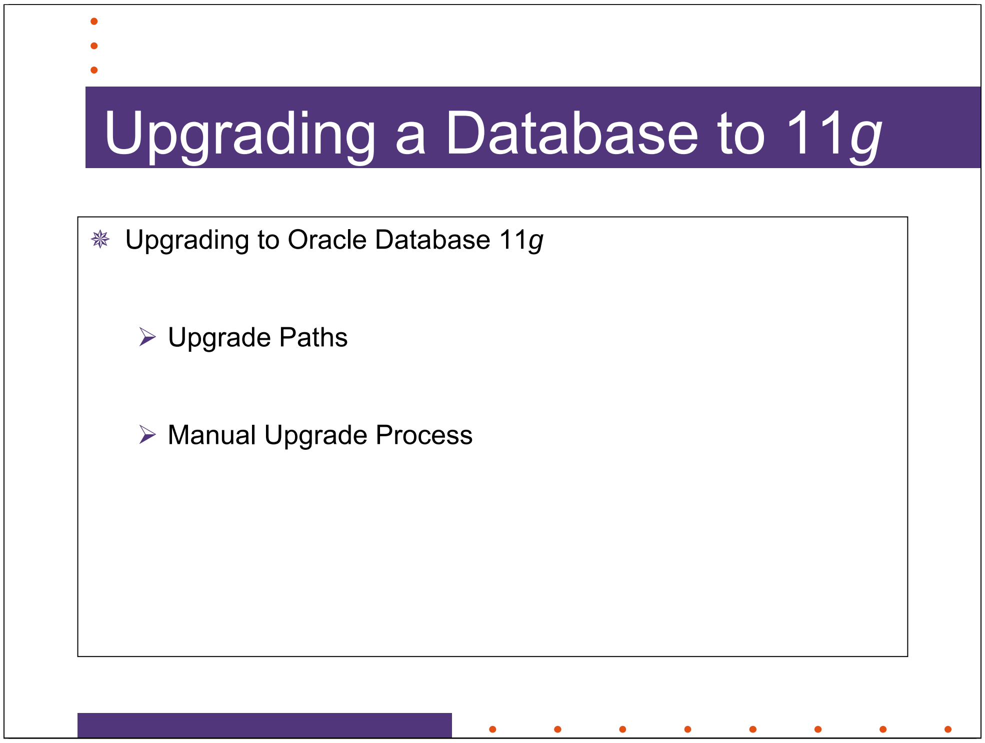
Lab 2: Using the DBCA (cont'd)

3. Use the 11*g* DBCA to create a new Oracle database.
   - Name this Oracle database and instance **o11g**.
   - Configure the database to use the Enterprise Manager.
   - Use the same password, *oracle*, for all accounts.
   - Specify a Flash Recovery area of 20G.
   - Enable Archiving.
   - Install the Sample Schemas
   - Use Automatic Memory Management, allocating a total of 250M of memory for this instance.
   - Generate Database Creation scripts; examine these scripts while the database is being created.
   - Carefully read the “Database creation complete” dialog.

   (Solution: *dbca1.txt*)

4. Examine the local *tnsnames.ora* file. If it lacks an entry for the new database, use the NETCA to create a Local Net Service Name for this database. (Solution: *tnsnames.txt*)

It is important to enable archiving, as later chapters include hot backups and media restore/recovery.



Upgrading a Database to 11*g*

- Upgrading to Oracle Database 11*g*
  - Upgrade Paths
  - Manual Upgrade Process

Text Page 17-24

Text Page 18 — Upgrading and the Compatibility Factor

Text Page 18 — Upgrade Path to Oracle 11*g*

Text Page 19-24 — Manual Upgrade Example

Manual Upgrade Process

compatible Parameter is not reversible.

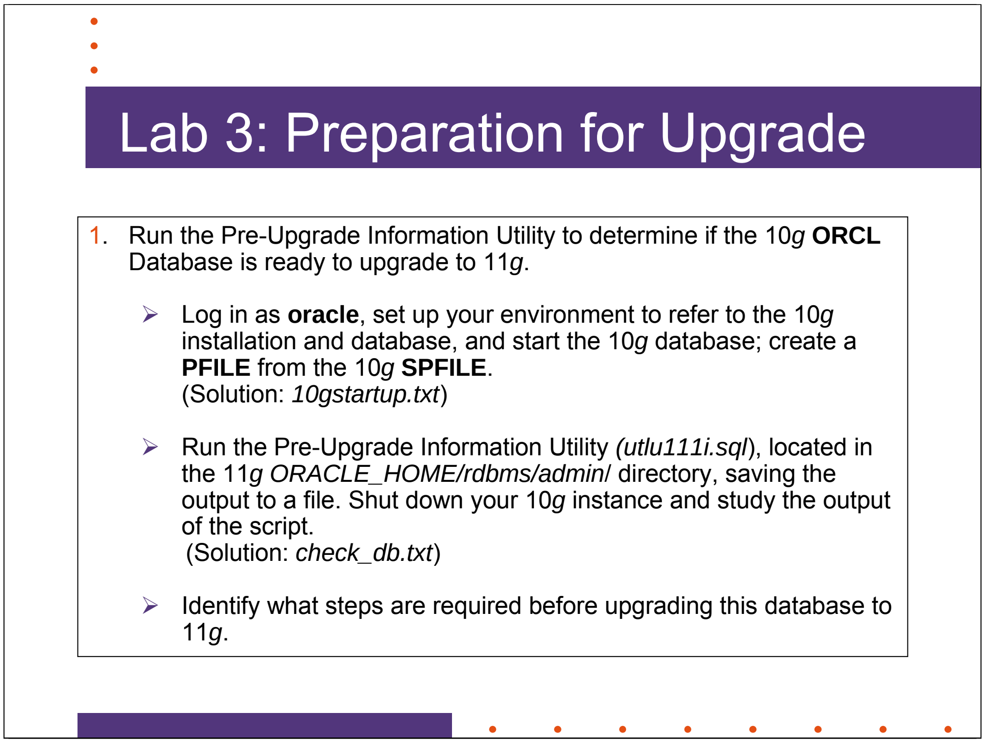
Lowest setting is 10.0.0.

Upgrade Scripts:

utlu111i — Pre-upgrade script

catupgrd — The upgrade script

utlu111s.sql — Post upgrade (status script)



Lab 3: Preparation for Upgrade

1. Run the Pre-Upgrade Information Utility to determine if the 10*g* **ORCL** Database is ready to upgrade to 11*g*.

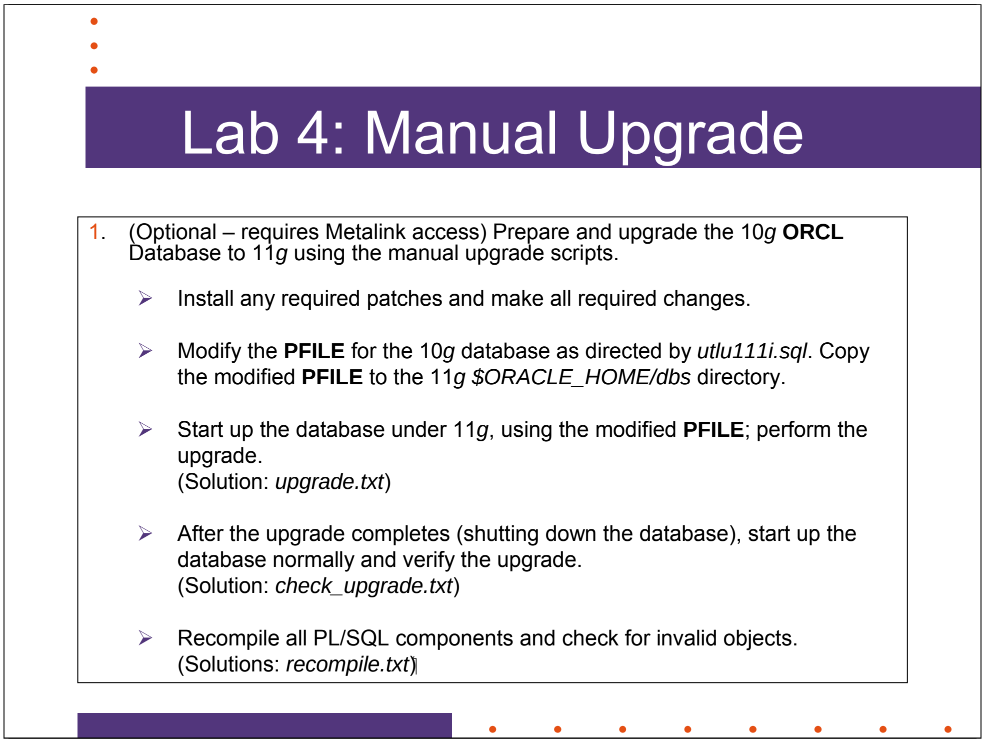
   - Log in as **oracle**, set up your environment to refer to the 10*g* installation and database, and start the 10*g* database; create a **PFILE** from the 10*g* **SPFILE**.
     (Solution: *10gstartup.txt*)

   - Run the Pre-Upgrade Information Utility *(utlu111i.sql*), located in the 11*g ORACLE_HOME/rdbms/admin/* directory, saving the output to a file. Shut down your 10*g* instance and study the output of the script.
     (Solution: *check_db.txt*)

   - Identify what steps are required before upgrading this database to 11*g*.

Students should have ***Upgrade Guide*** (in the ***Oracle Online Documentation***) open for use as the primary reference during this exercise.

Depending on the classroom hardware, it may be necessary to shut down the 11*g* instance before starting the 10*g* database.



Lab 4: Manual Upgrade

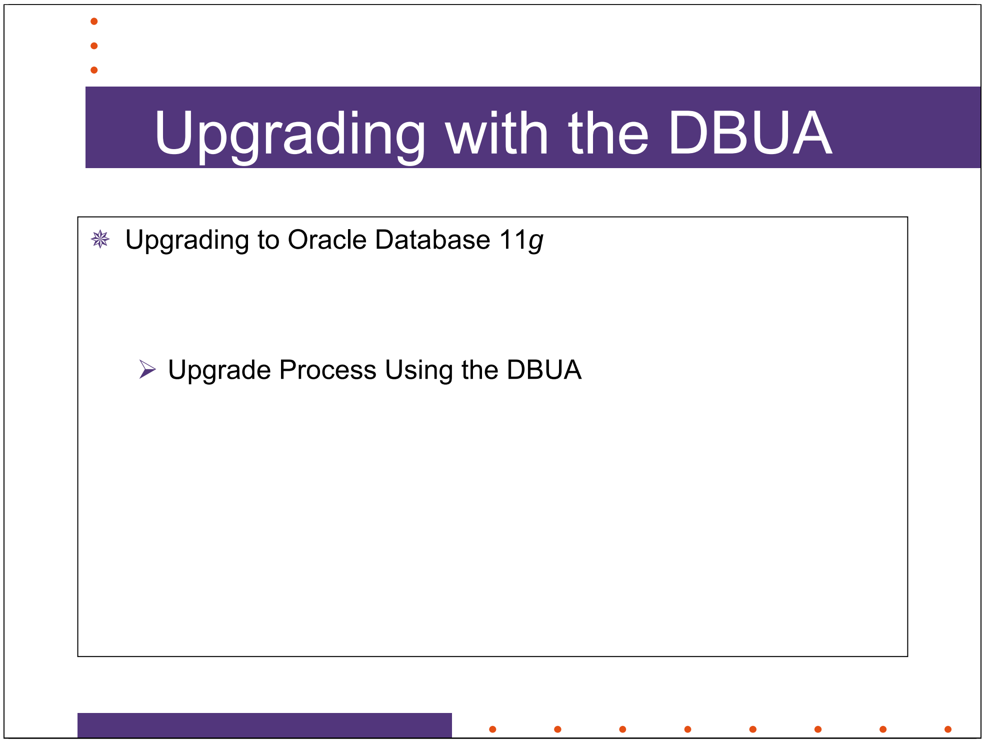
1. (Optional – requires Metalink access) Prepare and upgrade the 10*g* **ORCL** Database to 11*g* using the manual upgrade scripts.
   - Install any required patches and make all required changes.
   - Modify the **PFILE** for the 10*g* database as directed by *utlu111i.sql*. Copy the modified **PFILE** to the 11*g* *$ORACLE_HOME/dbs* directory.
   - Start up the database under 11*g*, using the modified **PFILE**; perform the upgrade.
     (Solution: *upgrade.txt*)
   - After the upgrade completes (shutting down the database), start up the database normally and verify the upgrade.
     (Solution: *check_upgrade.txt*)
   - Recompile all PL/SQL components and check for invalid objects.
     (Solutions: *recompile.txt*)

This lab is optional.

Fully preparing a 10*g* database for upgrade to 11*g* will likely require the student to obtain and install software patches available only via Oracle's Metalink paid support service. We cannot count on Metalink being accessible to all students. Also, the preparation and upgrade process can be involved and time-consuming, depending on what errors or issues arise.

Students should have ***Upgrade Guide*** (in the ***Oracle Online Documentation***) open for use as the primary reference during this exercise.

It is not necessary to complete an upgrade to continue with the course.



Upgrading with the DBUA

- Upgrading to Oracle Database 11*g*
  - Upgrade Process Using the DBUA

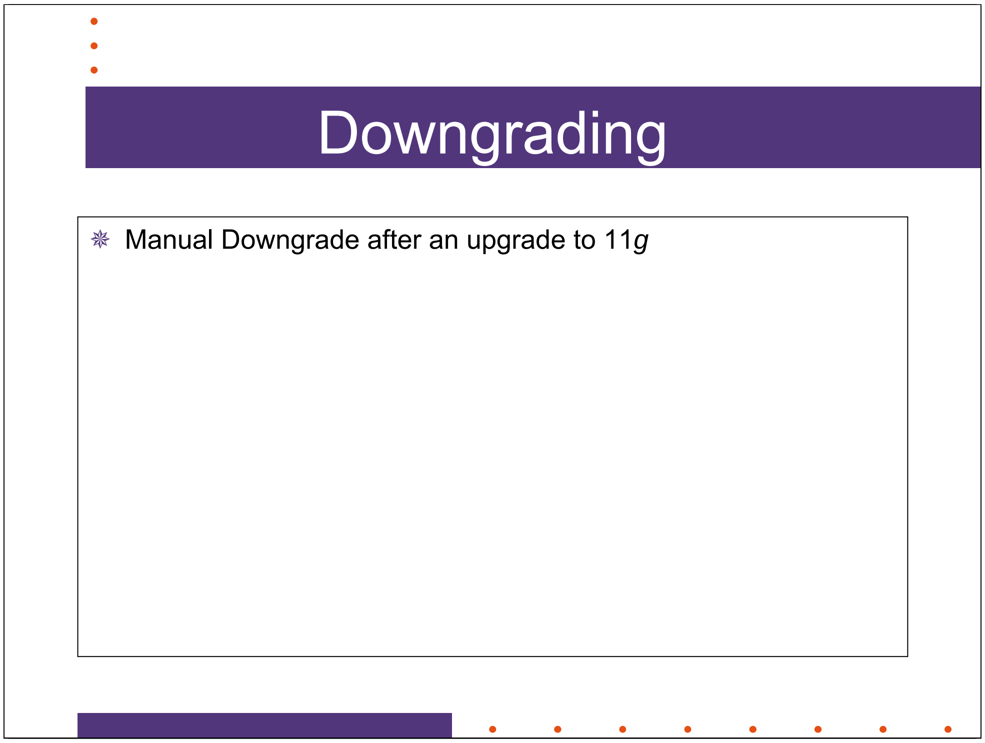
Text Page 24

Upgrade Process Using the DBUA

Invoking the DBUA

Backing Up Database Using the DBUA

Monitoring the Process



Downgrading

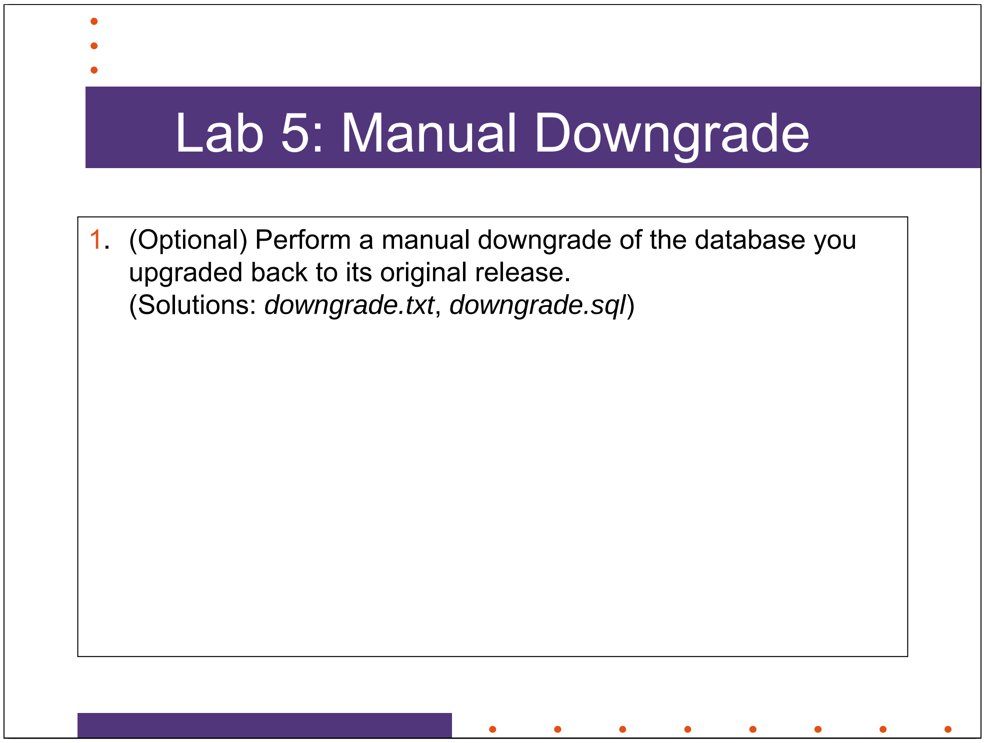
- Manual Downgrade after an upgrade to 11*g*

Text Page 25

A pre-11*g* database upgraded to 11*g* is downgradable.

Startup Downgrade option is needed.

catdwgrd.sql — Downgrade script

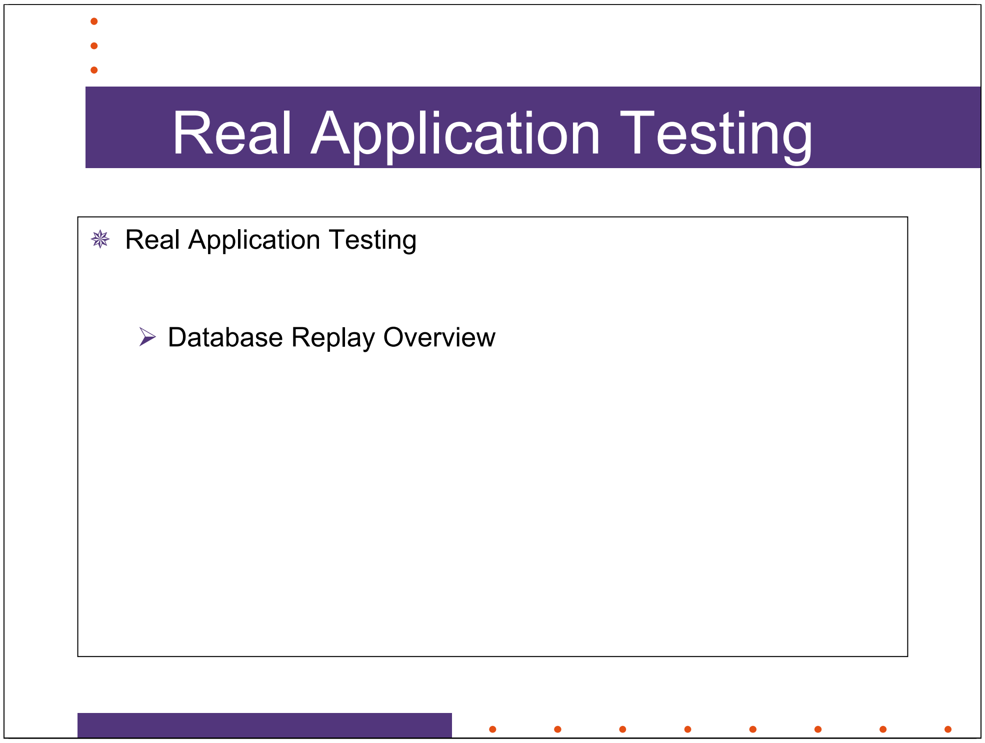


Lab 5: Manual Downgrade

1. (Optional) Perform a manual downgrade of the database you upgraded back to its original release.
   (Solutions: *downgrade.txt*, *downgrade.sql*)

This lab is only possible if you were able to successfully upgrade a database to 11*g* from a previous release. You cannot downgrade a database that was never upgraded, such as one originally created as 11*g*, to a lower version.

It is not necessary to complete this exercise to continue with class.



Real Application Testing

- Real Application Testing
  - Database Replay Overview

Text Page 35

Database Replay Overview

Replays actual workloads.

Allows you to test the effect of upgrades.

O/S

Database

Storage System Changes

Replay Steps:

1. Capture the existing workload.
2. Process the captured workload.
3. Replay the workload.
4. Analyze the replay.
5. Report the results.

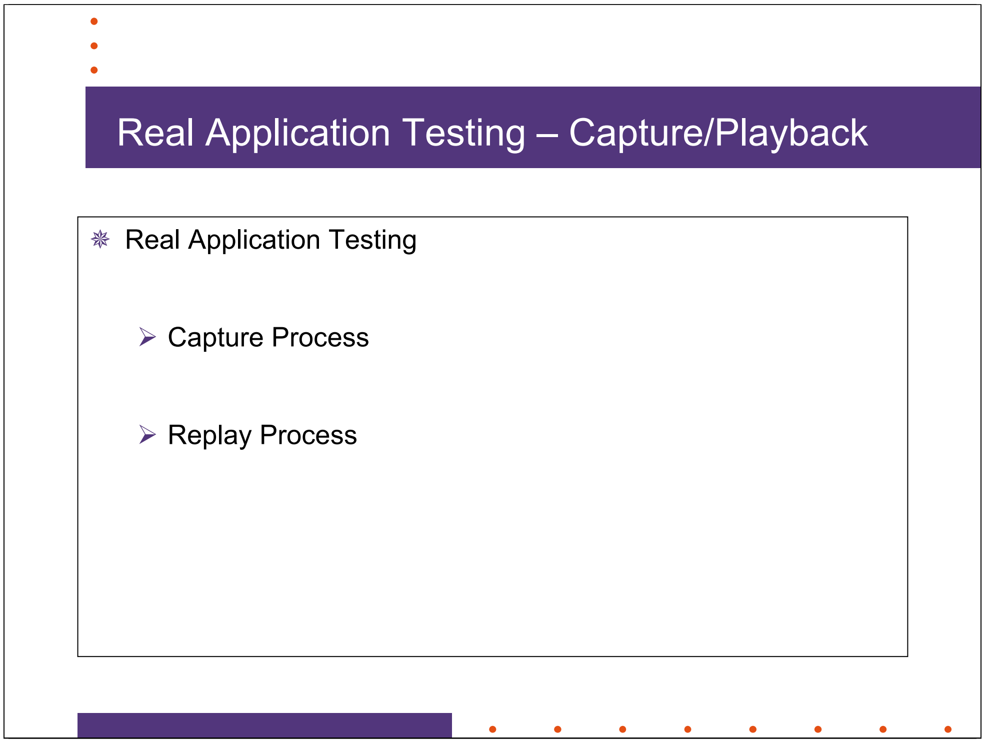
What is Captured:

DML statements

DDL statements

Session Control (Alter Session)

System Control Statements (Alter System)



Real Application Testing – Capture/Playback

- Real Application Testing
  - Capture Process
  - Replay Process

Text Page 39-46
Text Page 39 – Capturing the Workload
Text Page 42 – Replaying the Workload

Capture Process
Copy production system to another machine.
Startup restrict production system.
Create a directory object for the workload capture files.
Start the capture.

```
exec dbms_workload_capture.start_capture(name=>'firstcap', dir=>'testdir', duration => xxx);
```

Open production system for read write processing.
End the capture.

```
exec dbms_workload_capture.finish_capture();
```

Process the captured data.

```
exec dbms_workload_replay.process_capture(capture_dir=> 'testdir');
```

Replay Process
Replay the workload on test system.
Initialize the replay data.

```
exec dbms_workload_replay.initialize_replay(replay_name =>'test', replay_dir=> 'test_dir');
```

Loads data into replay tables on test system.
Start replay clients.
Use command-line interface start replay clients.
Calibrate the replay.

```
wrc userid/password mode=calibrate replay_dir=<fully qualified path to replay data>
```

Run the replay.

```
wrc userid/password mode=replay replay_dir=<fully qualified path to replay data>
```

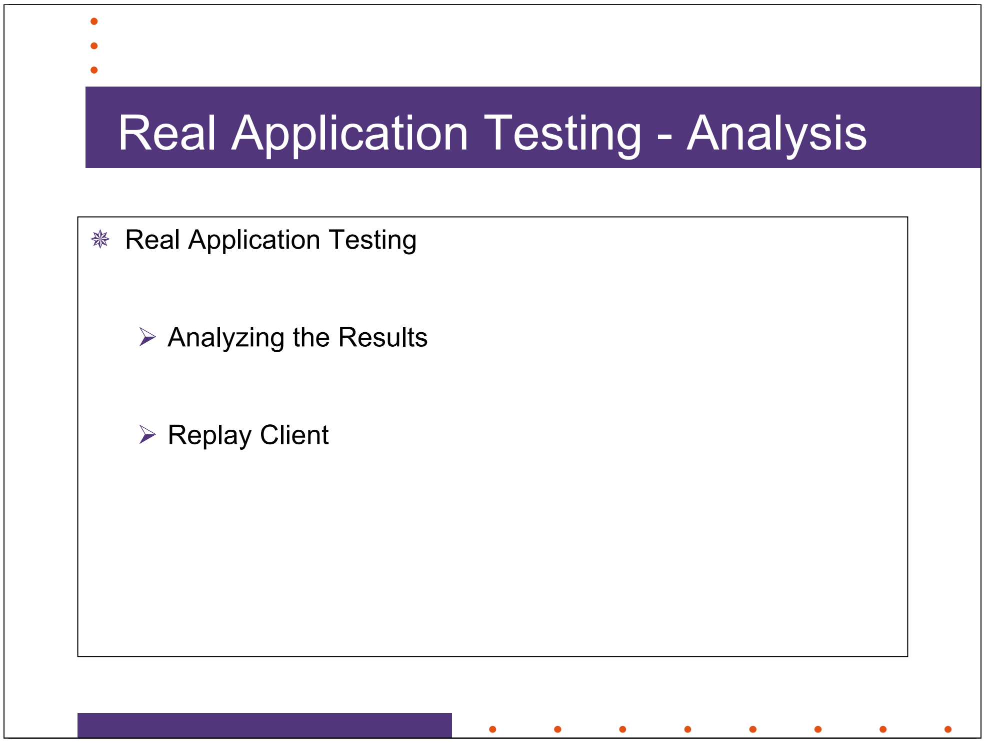
Start the replay processes.

```
exec dbms_workload_replay.prepare_replay(replay_name=> 'replay1', replay_dir=>'testdir',
    synchronization='FALSE');
exec dbms_workload_replay.start_replay();
```

Previously prepared clients begin processing replay data.

```
exec dbms_workload_replay.cancel_replay();
```

Stops prepared clients from processing replay data.



Real Application Testing - Analysis

- Real Application Testing
  - Analyzing the Results
  - Replay Client

Text Page 44

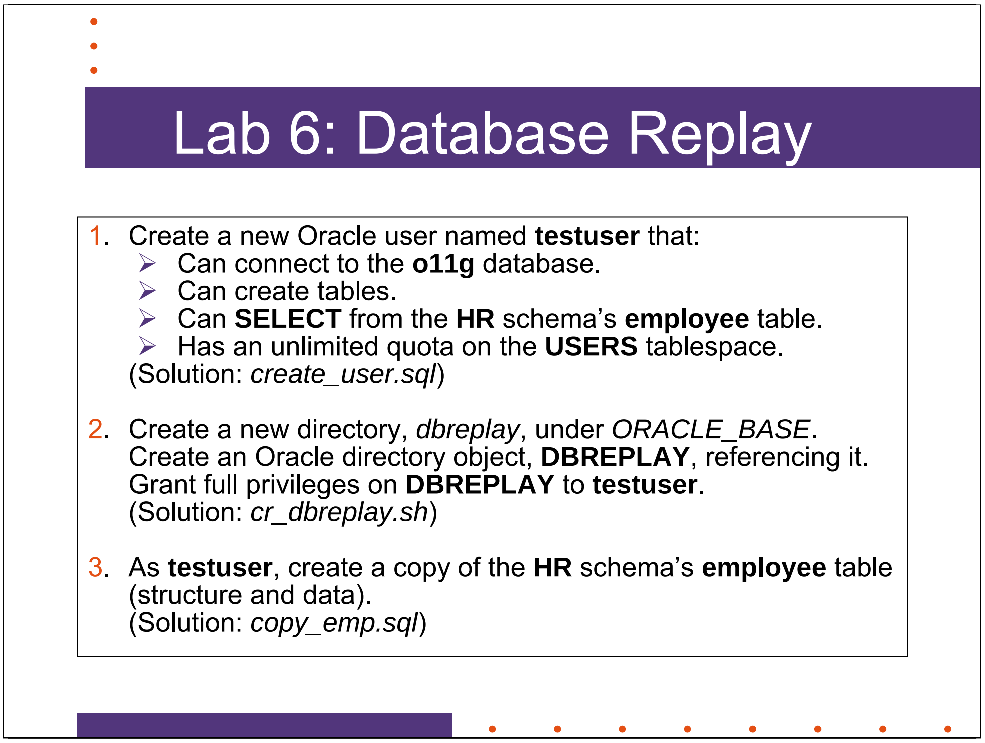
- Text Page 44 – Analyzing Workload Capture and Replay
- Text Page 43 – Replay Client

Analyzing the Results

- View **DBA_WORKLOAD_REPLAYS** will hold the data.
- Package/Procedure **dbms_workload_replay.get_replay_info** can provide the replay capture id.
- Using the id's you can, use the Package/Procedure **dbms_workload_replay.report** and output the results in text or HTML format.
- Report provides:
  - Error information that occurred during the replay.
  - Data Divergence.
    - Queries not returning the same results.
  - Replay Performance Metrics.
    - Change in processing time.
    - If greater processing time was needed, then this needs to be investigated prior to upgrade.

Replay Client

- Command-line tool
  - **wrc** (Workload Replay Client)
- SQL Performance Analyzer (SPA)



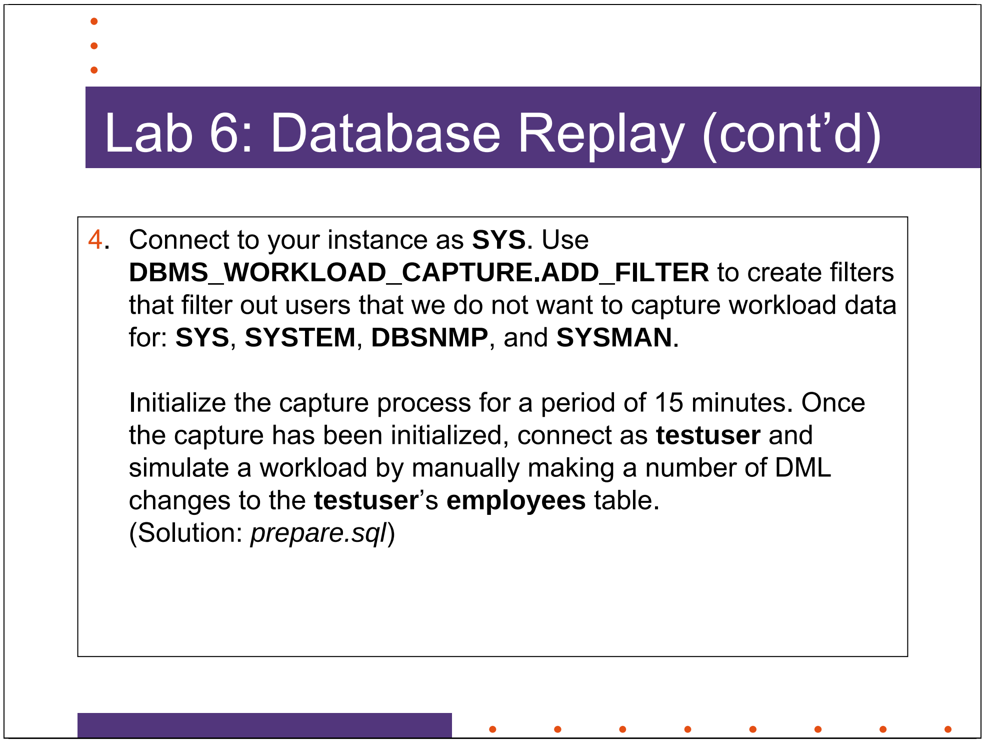
Lab 6: Database Replay

1. Create a new Oracle user named **testuser** that:
   - Can connect to the **o11g** database.
   - Can create tables.
   - Can **SELECT** from the **HR** schema’s **employee** table.
   - Has an unlimited quota on the **USERS** tablespace.

   (Solution: *create_user.sql*)

2. Create a new directory, *dbreplay*, under *ORACLE_BASE*. Create an Oracle directory object, **DBREPLAY**, referencing it. Grant full privileges on **DBREPLAY** to **testuser**.
   (Solution: *cr_dbreplay.sh*)

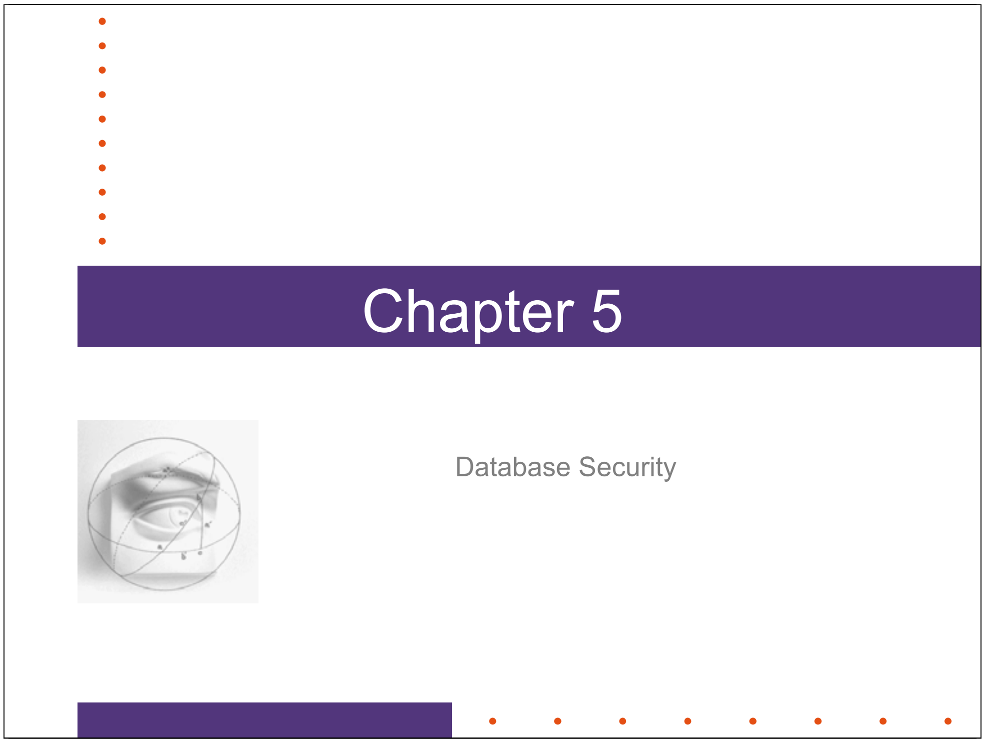
3. As **testuser**, create a copy of the **HR** schema’s **employee** table (structure and data).
   (Solution: *copy_emp.sql*)



Lab 6: Database Replay (cont'd)

4. Connect to your instance as **SYS**. Use **DBMS_WORKLOAD_CAPTURE.ADD_FILTER** to create filters that filter out users that we do not want to capture workload data for: **SYS**, **SYSTEM**, **DBSNMP**, and **SYSMAN**.

   Initialize the capture process for a period of 15 minutes. Once the capture has been initialized, connect as **testuser** and simulate a workload by manually making a number of DML changes to the **testuser**'s **employees** table.
   (Solution: *prepare.sql*)

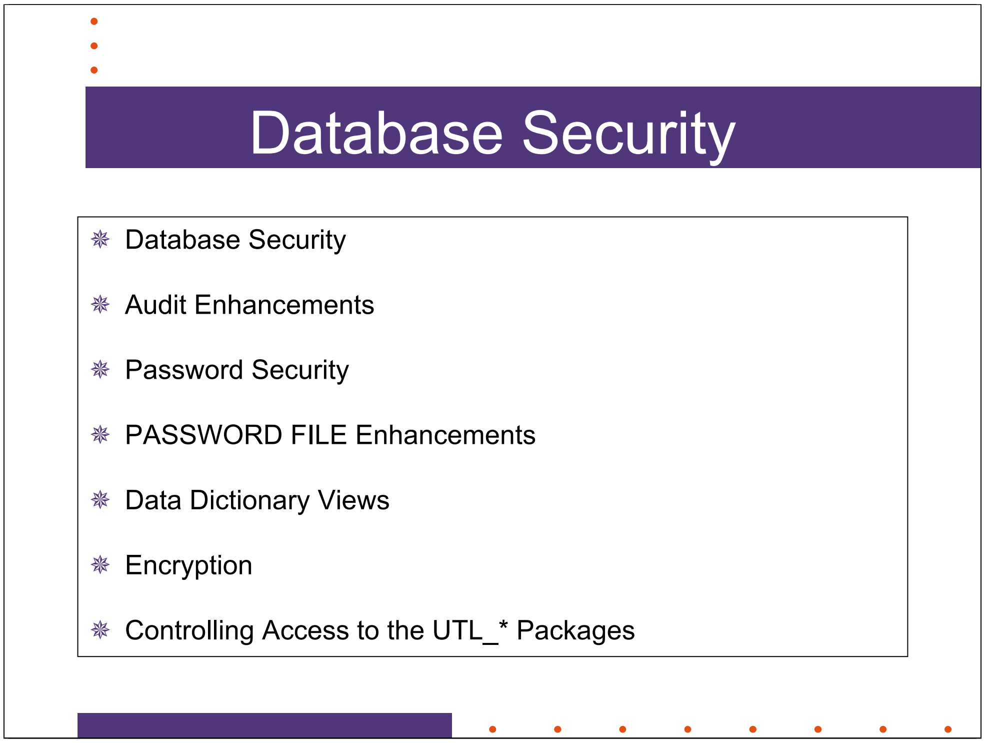


Chapter 5

Database Security

Students should open the following books in the Oracle Database Online Documentation 11*g*:

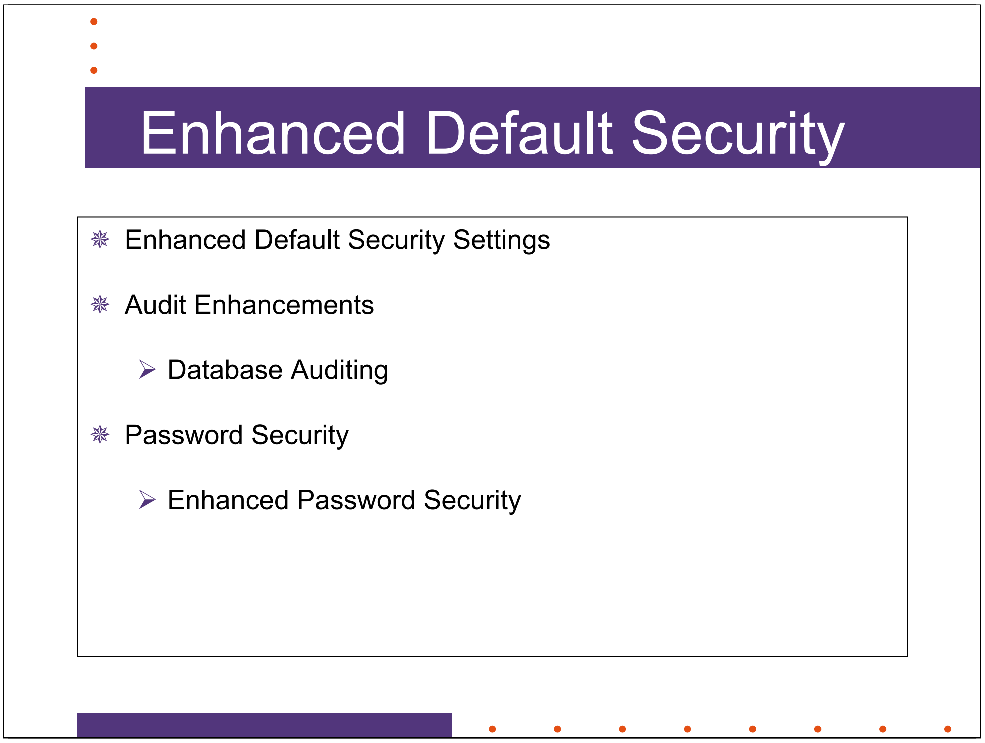
Advanced Security Administrator’s Guide



Database Security

- Database Security
- Audit Enhancements
- Password Security
- PASSWORD FILE Enhancements
- Data Dictionary Views
- Encryption
- Controlling Access to the UTL_* Packages

Text Page 225



Enhanced Default Security

- Enhanced Default Security Settings
- Audit Enhancements
  - Database Auditing
- Password Security
  - Enhanced Password Security

Text Page 225-239

- Text Page 225 – Database Security
- Text Page 260 – Advanced Security

Database Security

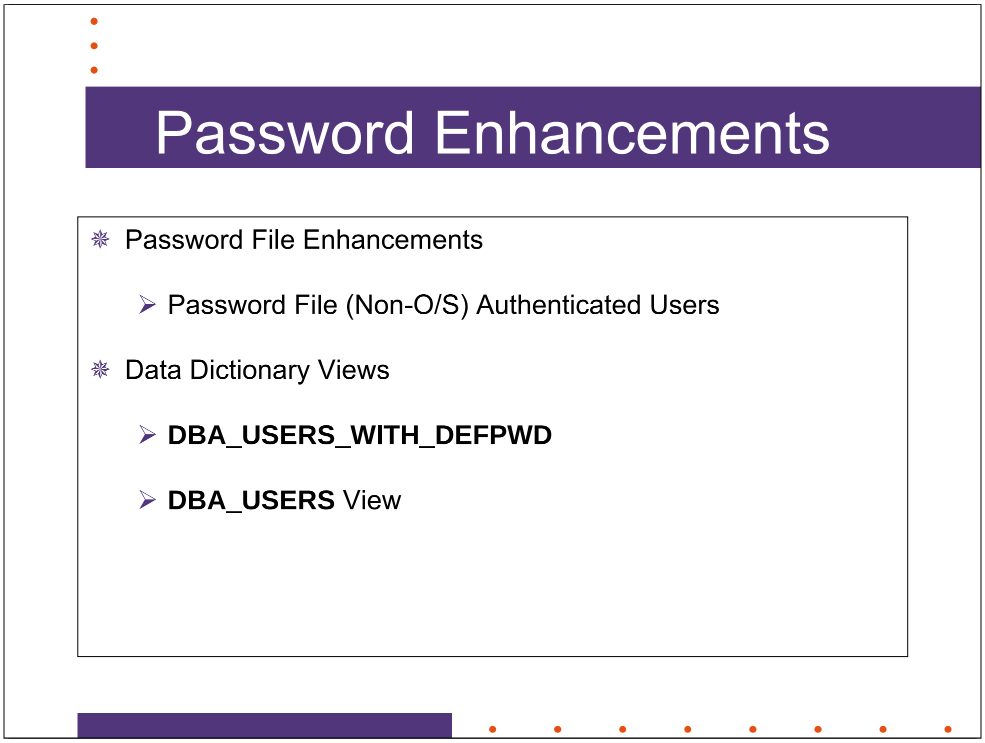
Database Auditing

- Database auditing enabled by default.
- Number of system level privileges are audited by default.
  - **dba_priv_audit_opts** view

Password Security

Enhanced Password Security

- SHA-1 encryption now used for passwords.
- Discourage brute force password cracking.
- New 11*g* password verification function.
- Case-sensitive passwords.
  - Not changed with upgrade.



Password Enhancements

- Password File Enhancements
  - Password File (Non-O/S) Authenticated Users
- Data Dictionary Views
  - **DBA_USERS_WITH_DEFPWD**
  - **DBA_USERS** View

Text Page 226-239

Text Page 261 – Password File Security

Password File (Non-O/S) Authenticated Users

Now Case Sensitive

DBA_USERS_WITH_DEFPWD

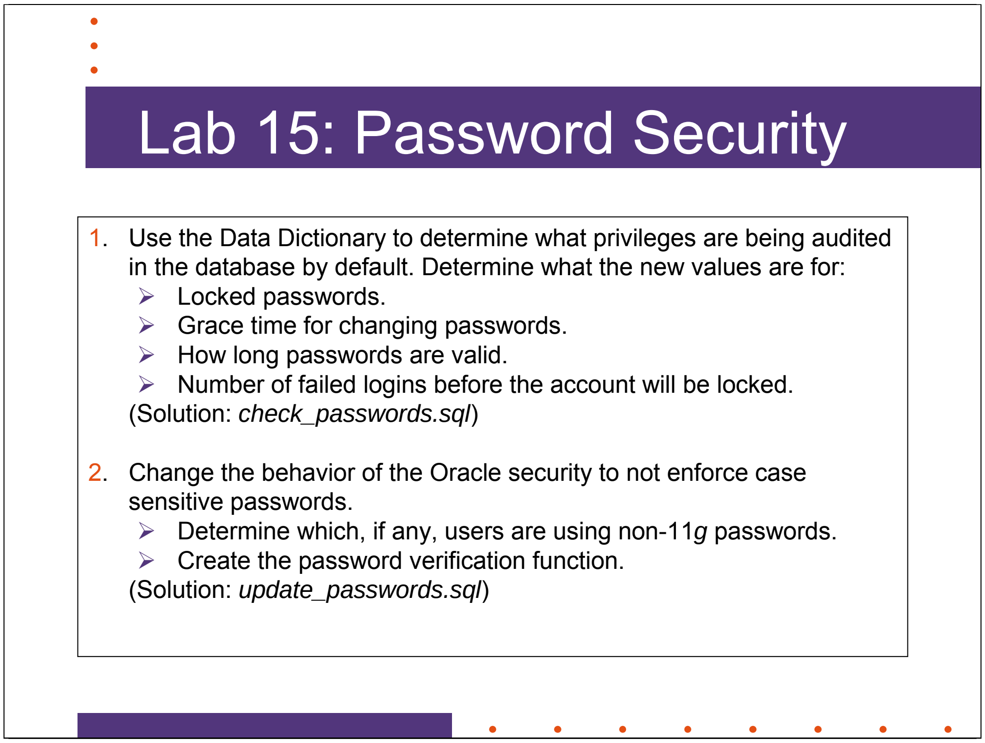
Catch those that have a default password set.

DBA_USERS View

password_versions

Stores the database password security being used.

Encrypted password no longer found in **DBA_USERS** view



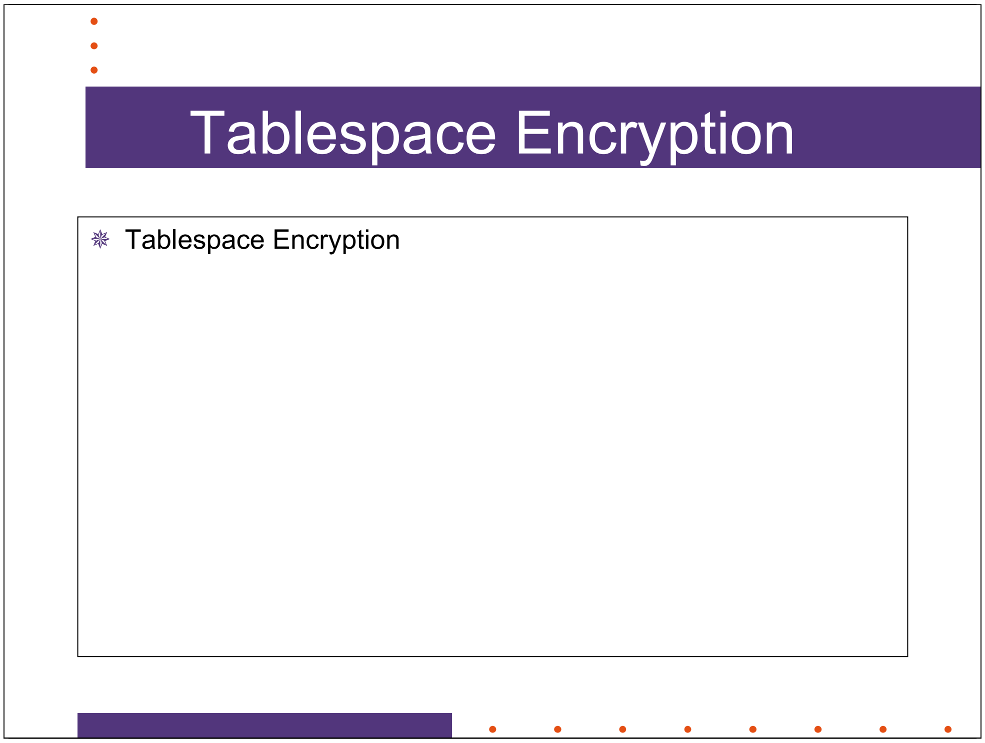
Lab 15: Password Security

1. Use the Data Dictionary to determine what privileges are being audited in the database by default. Determine what the new values are for:
   - Locked passwords.
   - Grace time for changing passwords.
   - How long passwords are valid.
   - Number of failed logins before the account will be locked.

   (Solution: *check_passwords.sql*)

2. Change the behavior of the Oracle security to not enforce case sensitive passwords.
   - Determine which, if any, users are using non-11*g* passwords.
   - Create the password verification function.

   (Solution: *update_passwords.sql*)



Tablespace Encryption

- Tablespace Encryption

Text Page 239-244

Tablespace Encryption

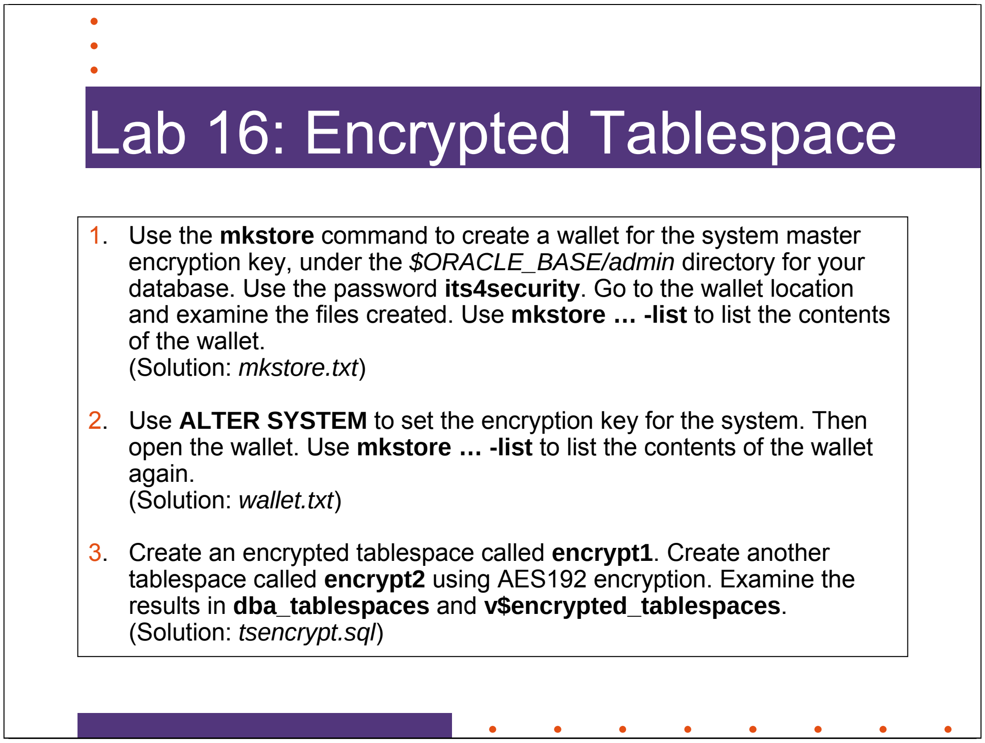
An entire tablespace can be encrypted in 11*g*.

Oracle Wallet used to store encryption keys.

Only new tablespaces can be encrypted.

Bfiles/External tables cannot be encrypted.

Temporary and Undo tablespaces cannot be encrypted.

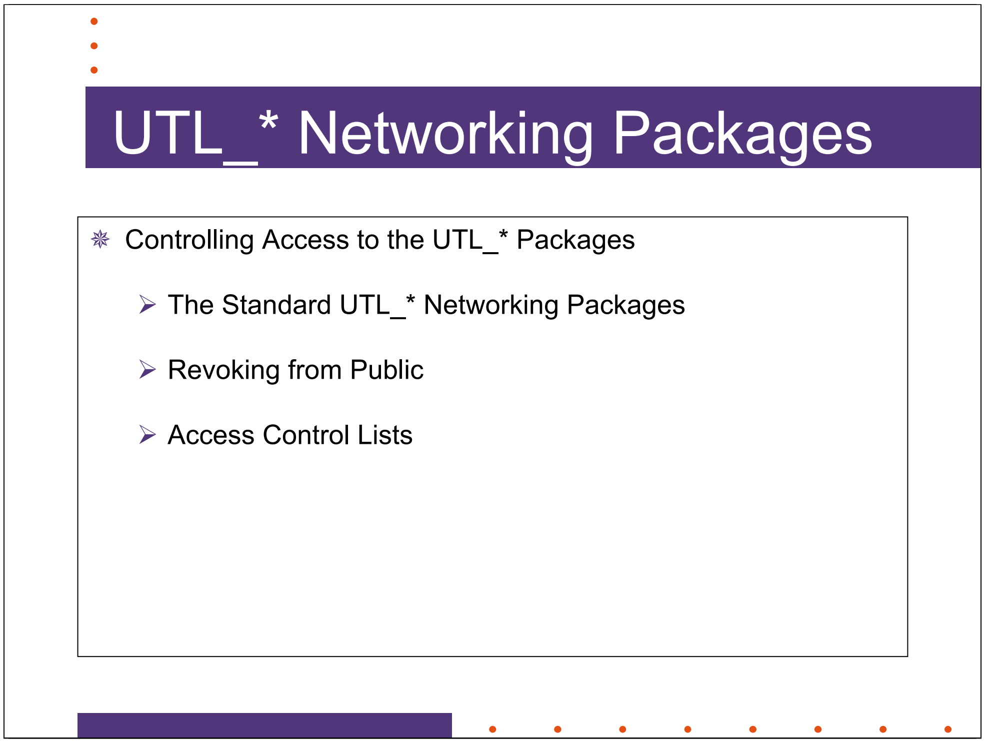


Lab 16: Encrypted Tablespace

1. Use the **mkstore** command to create a wallet for the system master encryption key, under the *$ORACLE_BASE/admin* directory for your database. Use the password **its4security**. Go to the wallet location and examine the files created. Use **mkstore … -list** to list the contents of the wallet.
   (Solution: *mkstore.txt*)

2. Use **ALTER SYSTEM** to set the encryption key for the system. Then open the wallet. Use **mkstore … -list** to list the contents of the wallet again.
   (Solution: *wallet.txt*)

3. Create an encrypted tablespace called **encrypt1**. Create another tablespace called **encrypt2** using AES192 encryption. Examine the results in **dba_tablespaces** and **v$encrypted_tablespaces**.
   (Solution: *tsencrypt.sql*)



UTL_* Networking Packages

- Controlling Access to the UTL_* Packages
  - The Standard UTL_* Networking Packages
  - Revoking from Public
  - Access Control Lists

Text Page 248-257

The Standard UTL* Packages

UTL_TCP

UTL_SMTP

UTL_MAIL

UTL_HTTP

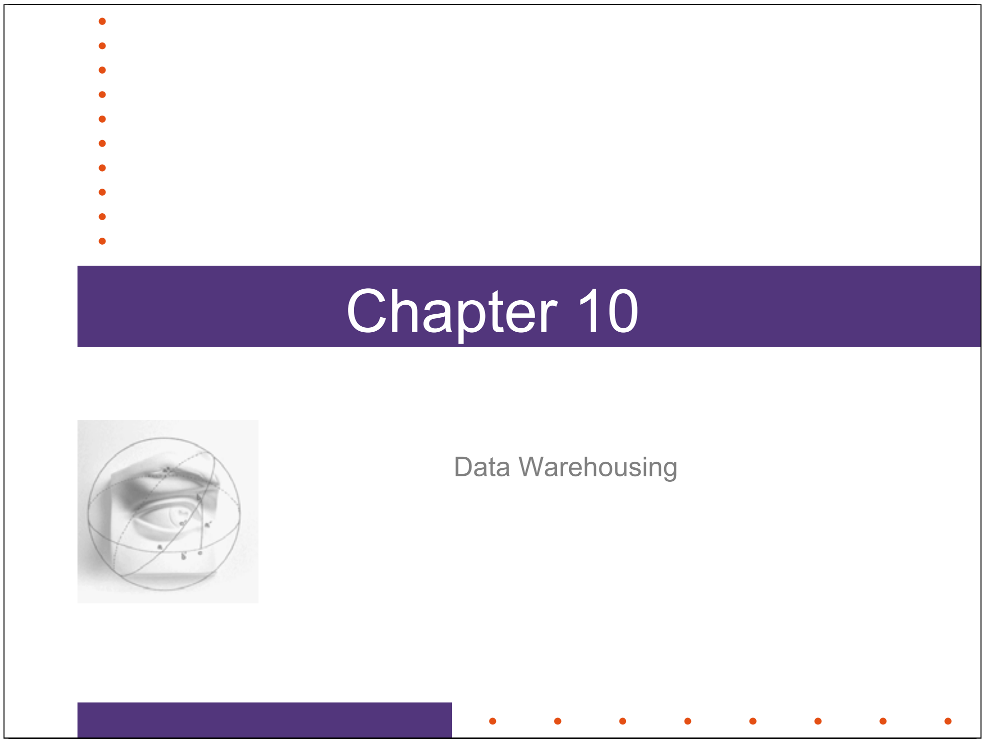
UTL_INADDR

Revoking from Public

Problems with blanket revoke from public.

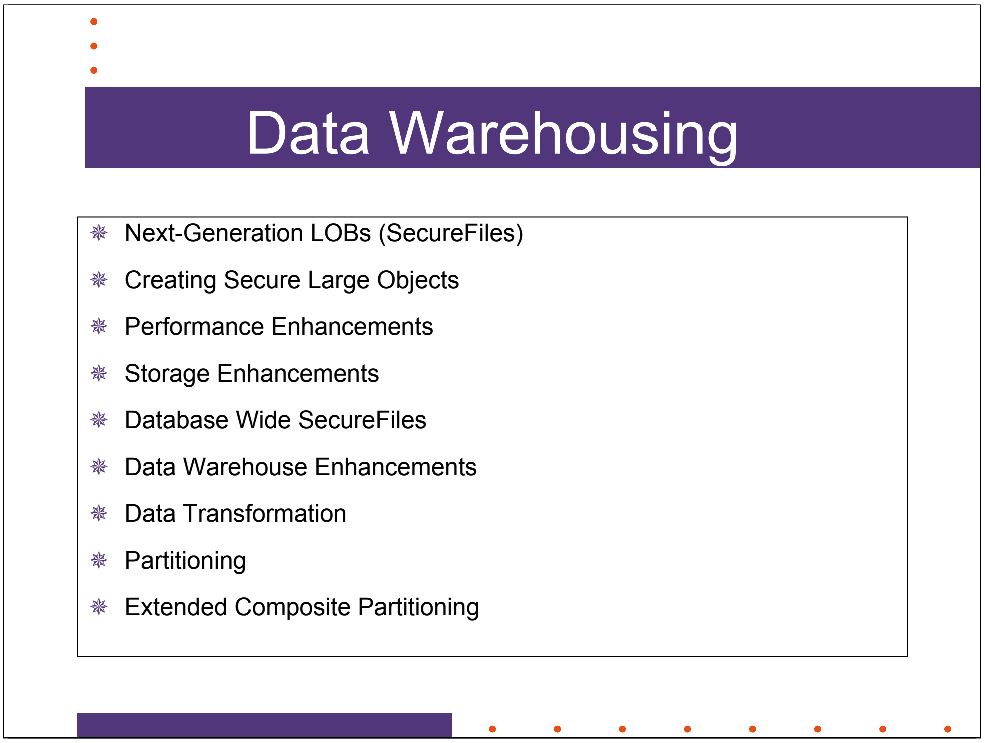
Access Control Lists

Fine-grained control



Chapter 10

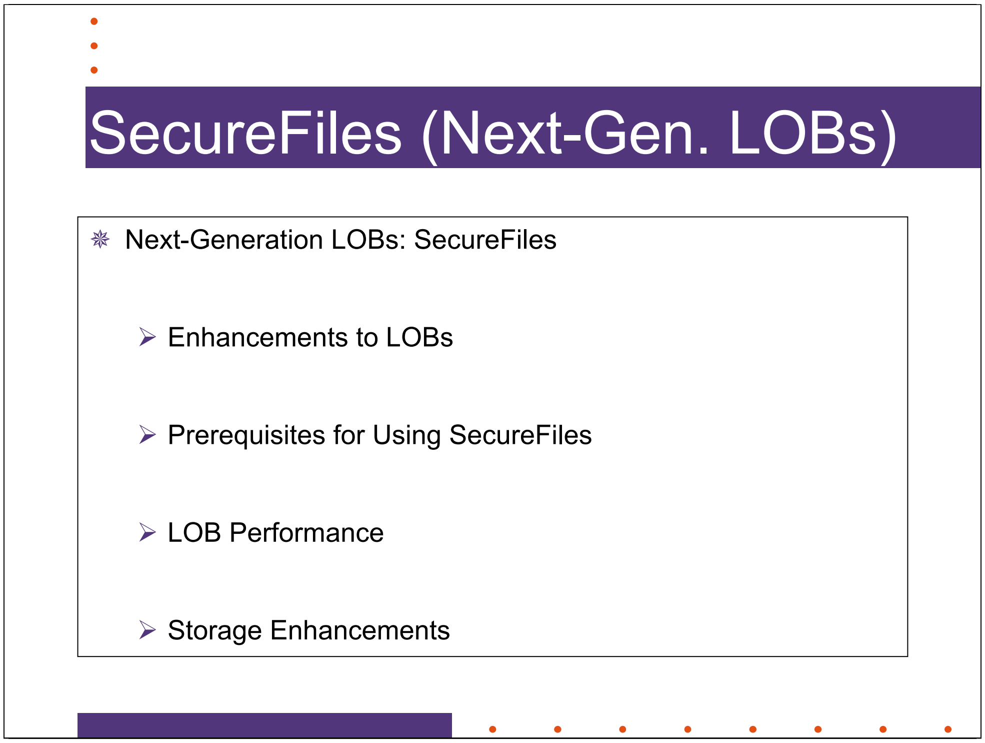
Data Warehousing



Data Warehousing

- Next-Generation LOBs (SecureFiles)
- Creating Secure Large Objects
- Performance Enhancements
- Storage Enhancements
- Database Wide SecureFiles
- Data Warehouse Enhancements
- Data Transformation
- Partitioning
- Extended Composite Partitioning

Text Page 515



SecureFiles (Next-Gen. LOBs)

- Next-Generation LOBs: SecureFiles
  - Enhancements to LOBs
  - Prerequisites for Using SecureFiles
  - LOB Performance
  - Storage Enhancements

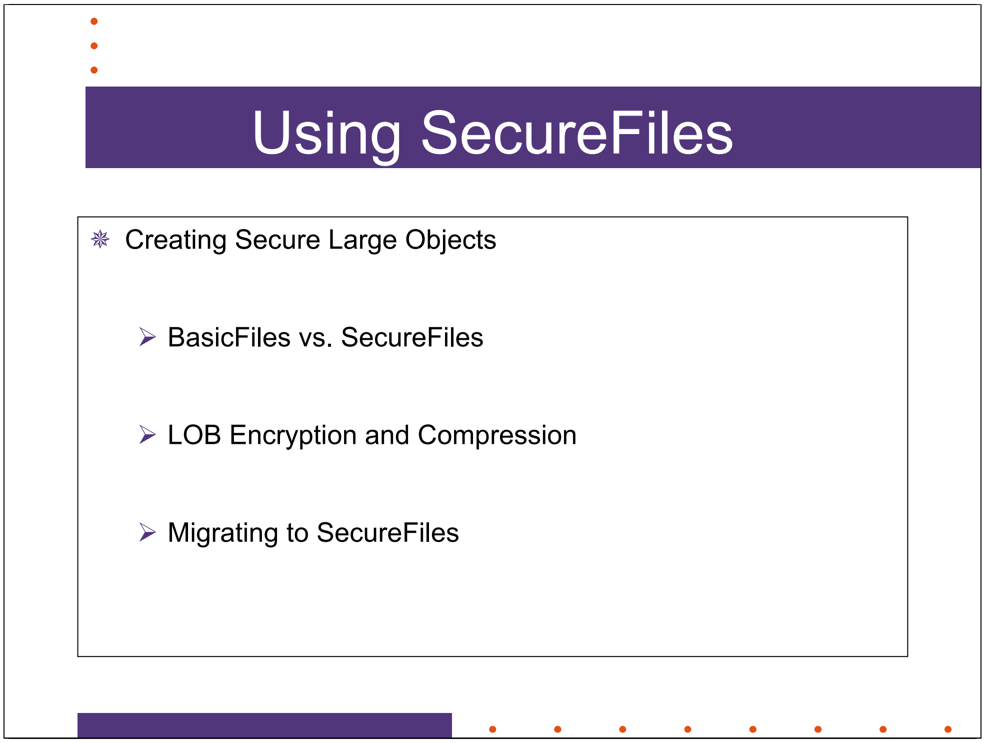
Text Page 516
Text Page 516 – Next Generation LOBS
Text Page 516 – Secure Files
Text Page 517 – Performance Enhancements
Text Page 518 – Storage Enhancements – Deduplication

Next-Generation LOBs (SecureFiles)
Improved Performance
Reduce Space Consumption
Enhanced Security

Prerequisites for using Secure Files
For compatibility, has to be 11.0.0.0.0 or higher.
Tablespace has to use ASSM.
Default starting with 10*g*.

LOB Performance
Allows for **cache**/**nocache** options.
RAC Environment:
New locking mechanism for LOBs to prevent cache-fusion lock contention.
Writes of LOBs
Option for Metadata logging with SecureFiles
Reduces the mean time to recover for LOBs.

Storage Enhancements
Deduplication Option:
Reduces duplicate data in a LOB, reducing storage requirements.
Add deduplication demo here.



Using SecureFiles

- Creating Secure Large Objects
  - BasicFiles vs. SecureFiles
  - LOB Encryption and Compression
  - Migrating to SecureFiles

Text Page 516-527

Text Page 517 — Creating Secure Files

Text Page 522 — LOB Encryption

Text Page 523 — LOB Compression

Text Page 526 — Migrating to SecureFiles

Creating Secure Large Objects

BasicFiles vs. SecureFiles

New option when creating a LOB.

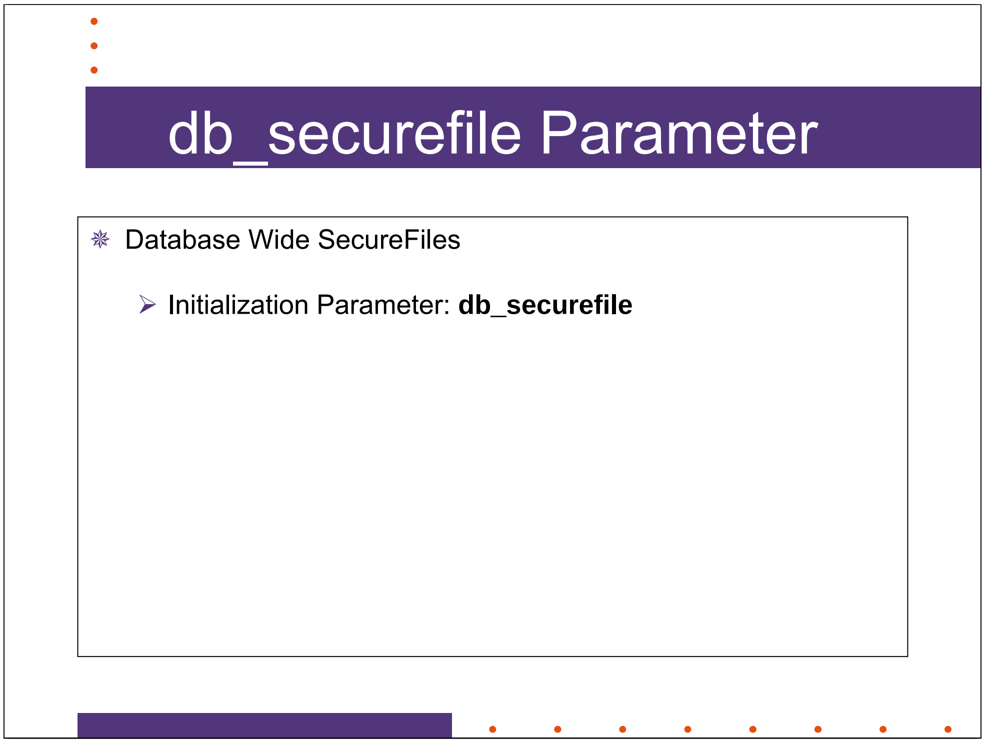
A storage option to the **TABLE** clause.

Example of a SecureFile:

```
CREATE TABLE secure_docs
    (document_id NUMBER
    ...
    blob_cotent BLOB)
    TABLESPACE tools
    LOB (blob_content) STORE AS SECUREFILE)
```

Example of a BasicFile:

```
CREATE TABLE secure_docs
    (document_id NUMBER
    ...
    blob_cotent BLOB)
    TABLESPACE tools
    LOB (blob_content) STORE AS BASICFILE)
```



db_securefile Parameter

- Database Wide SecureFiles
  - Initialization Parameter: **db_securefile**

Text Page 525-526

Text Page 525 – **db_securefile** Parameter

New Parameter **db_securefile**

Values are:

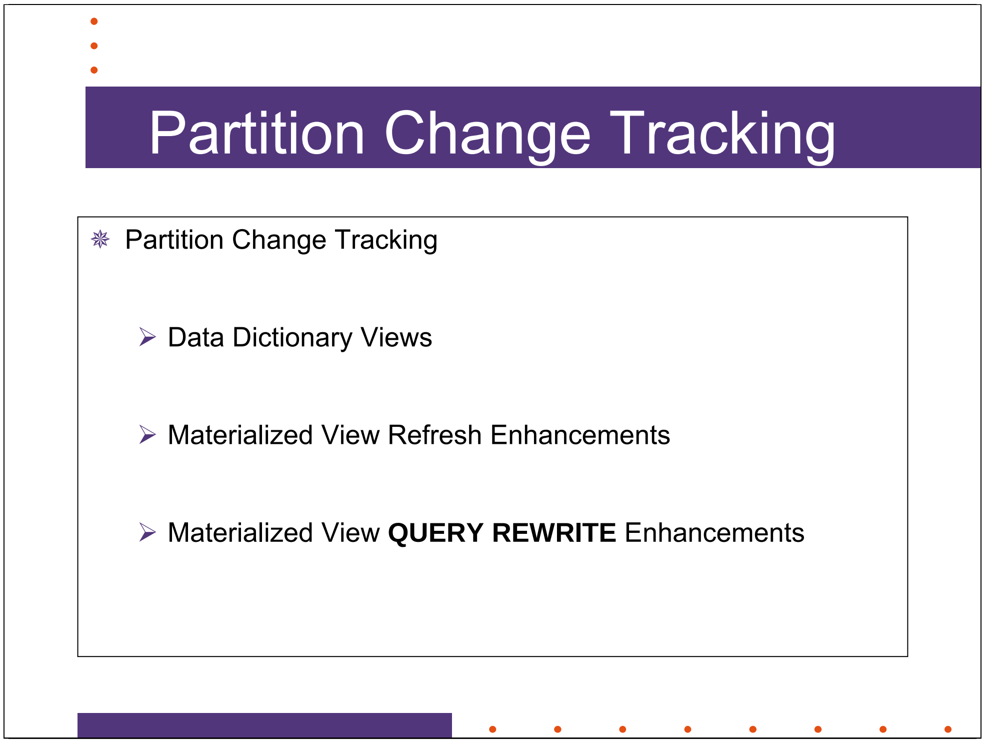
always – Creates LOBs as SecureFiles regardless of parameters specified.
Must be an ASSM TS.

force – Forces all tables that have LOBs to be SecureFiles.
Tablespace must be ASSM, otherwise an error is returned.

permitted – Default setting; SecureFiles are created on a per-table basis.

never – No SecureFiles can be created.

ignore – Disallows SecureFiles; ignores storage options.



Partition Change Tracking

- Partition Change Tracking
  - Data Dictionary Views
  - Materialized View Refresh Enhancements
  - Materialized View **QUERY REWRITE** Enhancements

Text Page 527-532

- Text Page 527 — Data Warehousing New Features
- Text Page 528 — Data Dictionary Views
- Text Page 529 — Materialized View Enhancements

Data Warehouse Enhancements

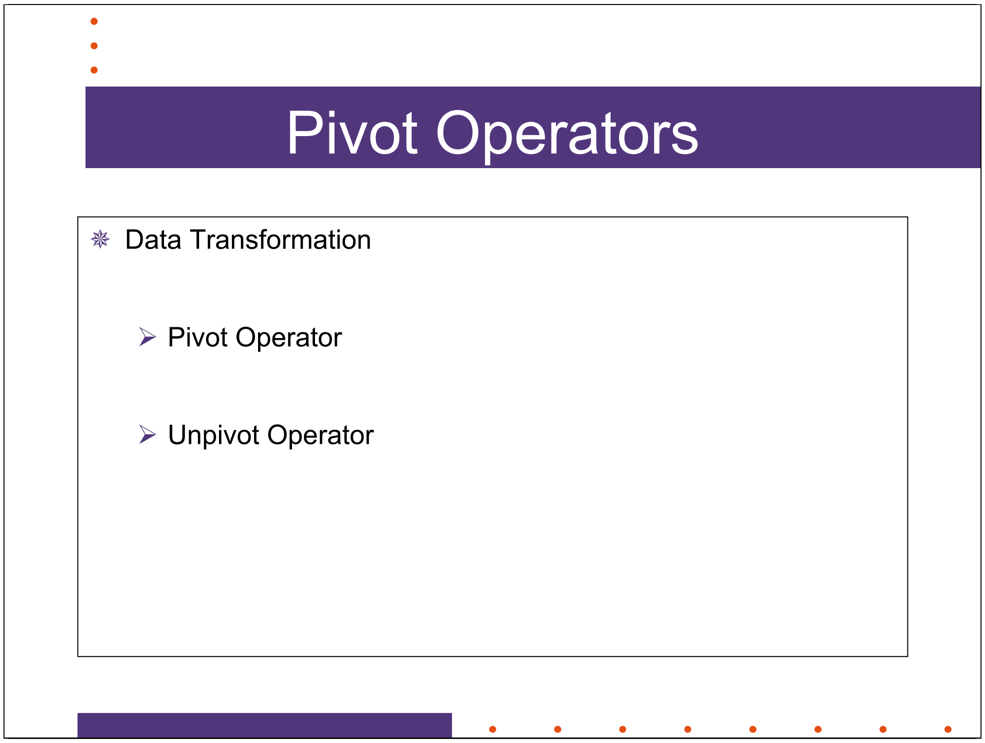
Data Dictionary Views for Partition Freshness

- Partition Change Tracking
  - Introduced in 9*i*, now have Data Dictionary Views.
  - **ALL_MVIEWS**
  - **ALL_MVIEW_DETAIL_RELATIONS**
  - **ALL_MVIEW_DETAIL_PARTITION**
  - **ALL_MVIEW_DETAIL_SUBPARTITION**

Materialized View Refresh Performance Improvements

Materialized View **QUERY REWRITE** Enhancements

- Typically a rewrite will occur if:
  - Select columns are the same.
  - **FROM** clause has to reference same objects.
  - **HAVING** clause has to be the same.
  - Join conditions including **WHERE** clause are equivalent.
  - Inline views must be the same.
- Syntax of inline views no longer has to match exactly.



Pivot Operators

- Data Transformation
  - Pivot Operator
  - Unpivot Operator

Text Page 532-538

Text Page – 532 Data Transformation

Text Page – 532 Pivot Operator

Text Page – 537 Unpivot Operator

Data Transformation

Pivot Operator

New operator to replace case logic to pivot data into correct category.

Example:

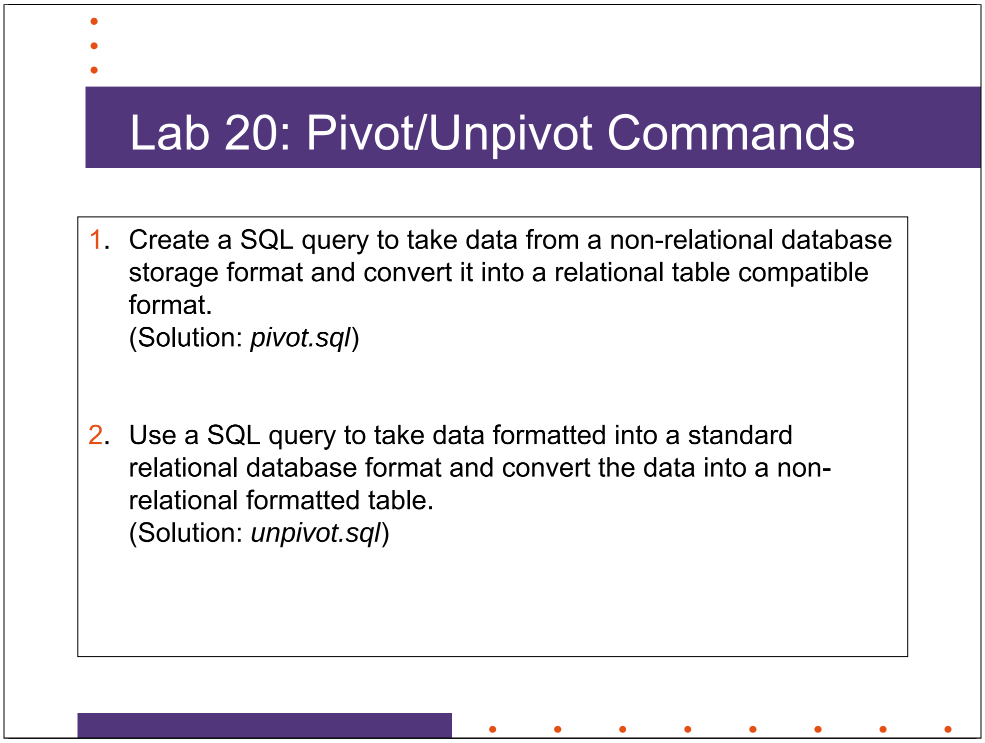
```
SELECT video_name ,
      SUM(CASE WHEN month  ='01' THEN quantity_rented ELSE NULL END) jan,
      SUM(CASE WHEN month = '02' tTHEN quantity_rented ELSE NULL END) feb,
      ...
      FROM (SELECT video_name, month, quantity_rented FROM video_mstr_vw)
      GROUP BY video_name
      ORDER BY 1
```

Pivot Syntax:

```
SELECT *
      FROM SELECT video_name, month, quantity_rented from video_mstr_vw
      PIVOT (SUM (quantity_rented) FOR month IN
            ('01' AS jan,
            '02' AS feb,
            ...
            '12' AS dec))
      ORDER BY video_name DESC;
```

Unpivot Operator

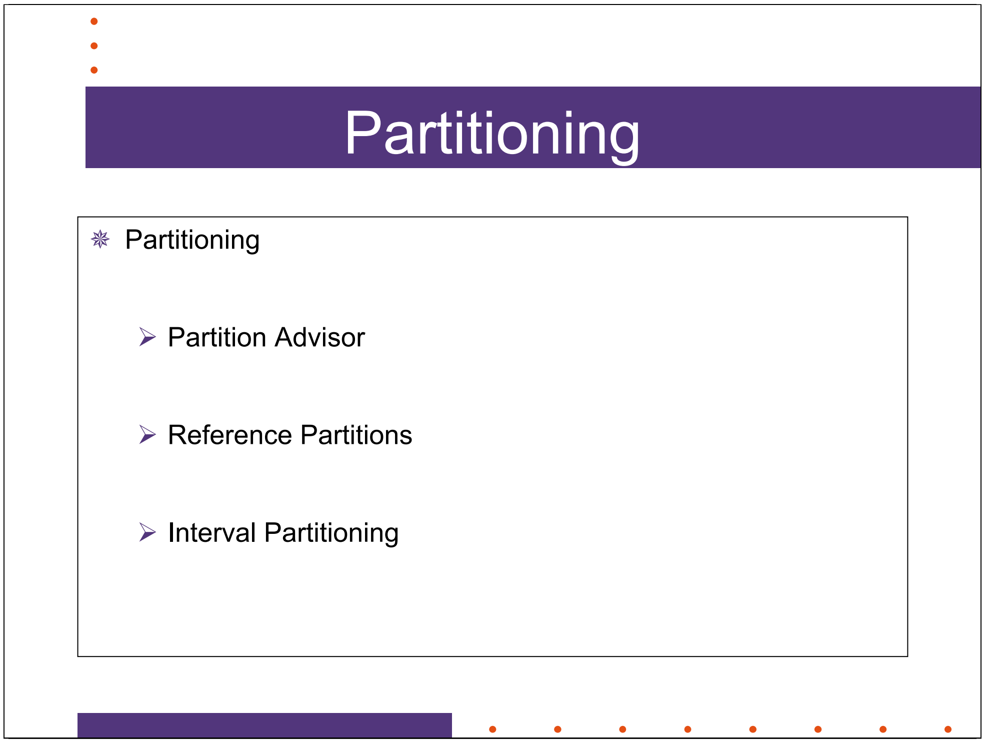
Rotates data from columns into rows.



Lab 20: Pivot/Unpivot Commands

1. Create a SQL query to take data from a non-relational database storage format and convert it into a relational table compatible format.
   (Solution: *pivot.sql*)

2. Use a SQL query to take data formatted into a standard relational database format and convert the data into a non-relational formatted table.
   (Solution: *unpivot.sql*)



Partitioning

- Partitioning
  - Partition Advisor
  - Reference Partitions
  - Interval Partitioning

Text Page 538-552

Text Page 538 ─ Partitioning

Text Page 539 ─ Partition Advisor

Text Page 544 ─ Reference Partition

Text Page 546 ─ Interval Partition

Partitioning

Partition Advisor

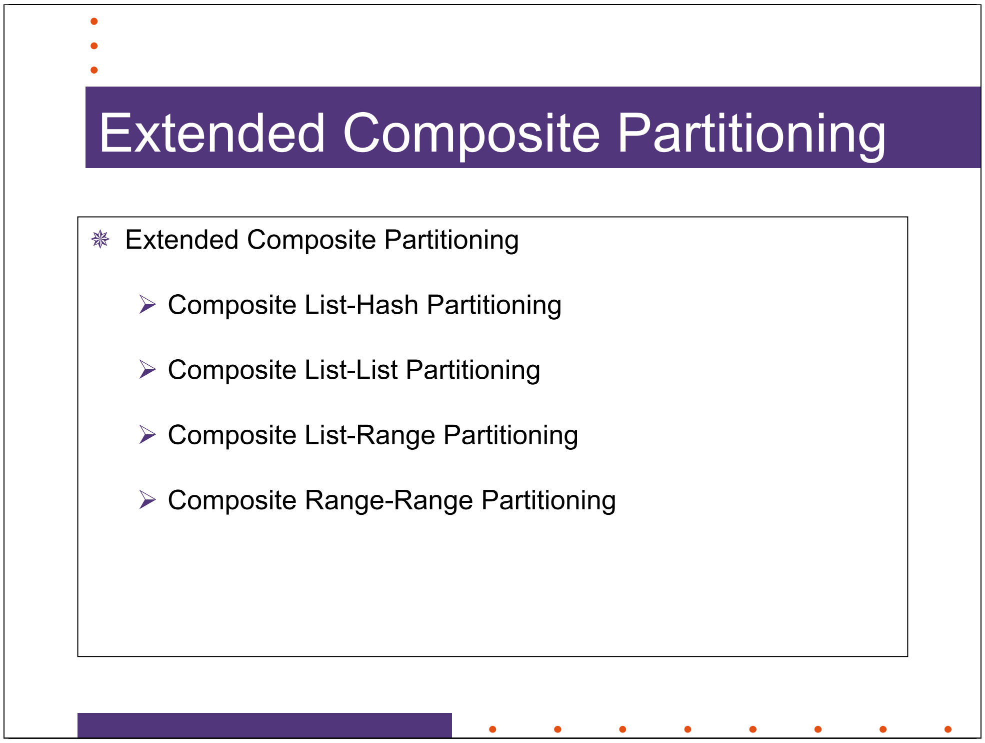
SQL Access Advisor ─ Now recommends partitioning as an option to improve access paths to data.
Partition Advisor will provide the appropriate SQL syntax to create a partitioned object out of a non partitioned object.
Part of the Oracle Tuning Pack.
Demo of screen shots for partitioning.

Reference Partitions

Based on the Primary Key-Foreign Key relationship between two tables.
Child table automatically inherits the partitioning key from the parent table.
Demo of code to create a Reference Partition.

Interval Partitioning

For partitioned data that has to be added on a weekly/monthly/quarterly basis.
Extension of range partitions.
Demo of code to create an interval partition table.



Extended Composite Partitioning

- Extended Composite Partitioning
  - Composite List-Hash Partitioning
  - Composite List-List Partitioning
  - Composite List-Range Partitioning
  - Composite Range-Range Partitioning

Text Page 552-556

Text Page 552 — Composite List-Hash Partitioning

Text Page 553 — Composite List-List Partitioning

Text Page 553 — Composite List-Range Partitioning

Text Page 554 — Composite Range-Range Partitioning

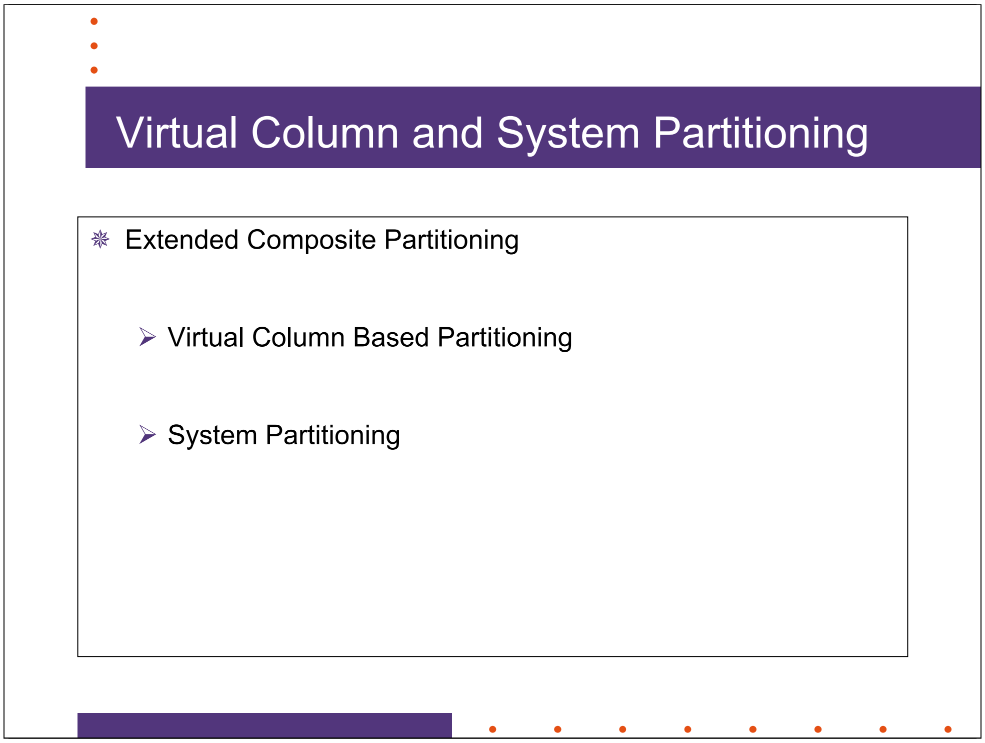
Extended Composite Partitioning
Four new extensions to the partitioning scheme:
List-Hash
List-List
List-Range
Range-Range

Composite List-Hash Partitioning
List partition first followed by hash partition.

Composite List-List Partitioning
List partition first followed by list partition.
Sample code of list-list partition.

Composite List-Range Partitioning
List partition first followed by range partition.
Sample code of list-range partition.

Composite Range-Range Partitioning
Range partition followed by range partition
Sample code of range-range partition.



Virtual Column and System Partitioning

- Extended Composite Partitioning
  - Virtual Column Based Partitioning
  - System Partitioning

Text Page 556-561

Text Page 556 – Virtual Column Partitioning

Text Page 559 – System Partitioning

Virtual Column Based Partitioning

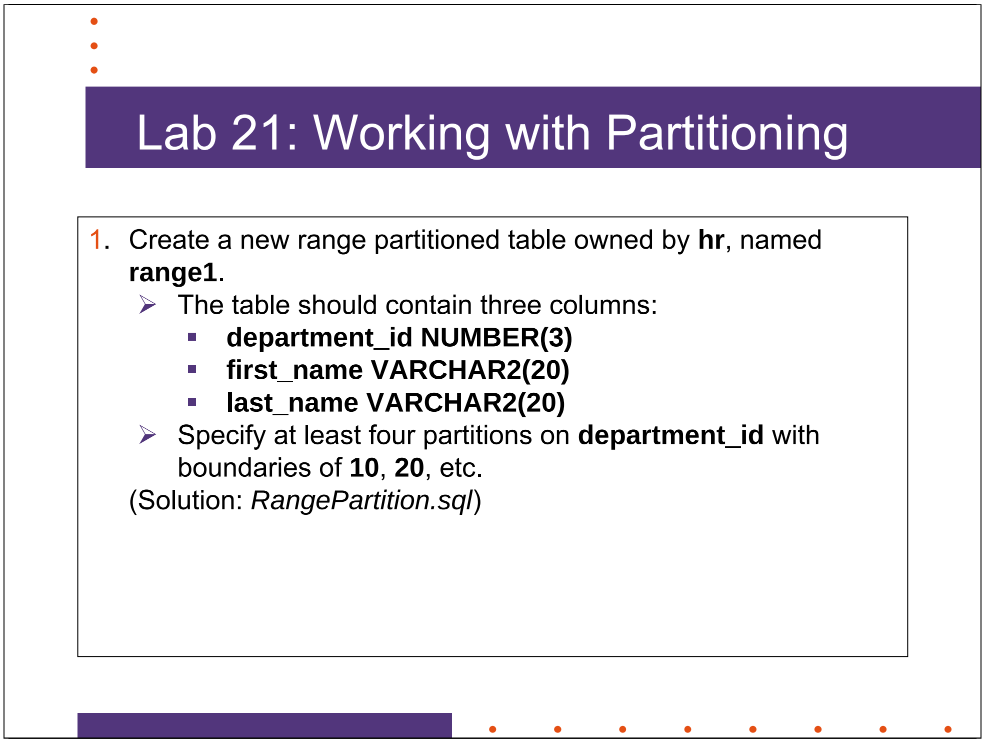
Partitioning by virtual columns.

Virtual columns are columns that are based on calculations.

System Partitioning

Application code determines partitioning.

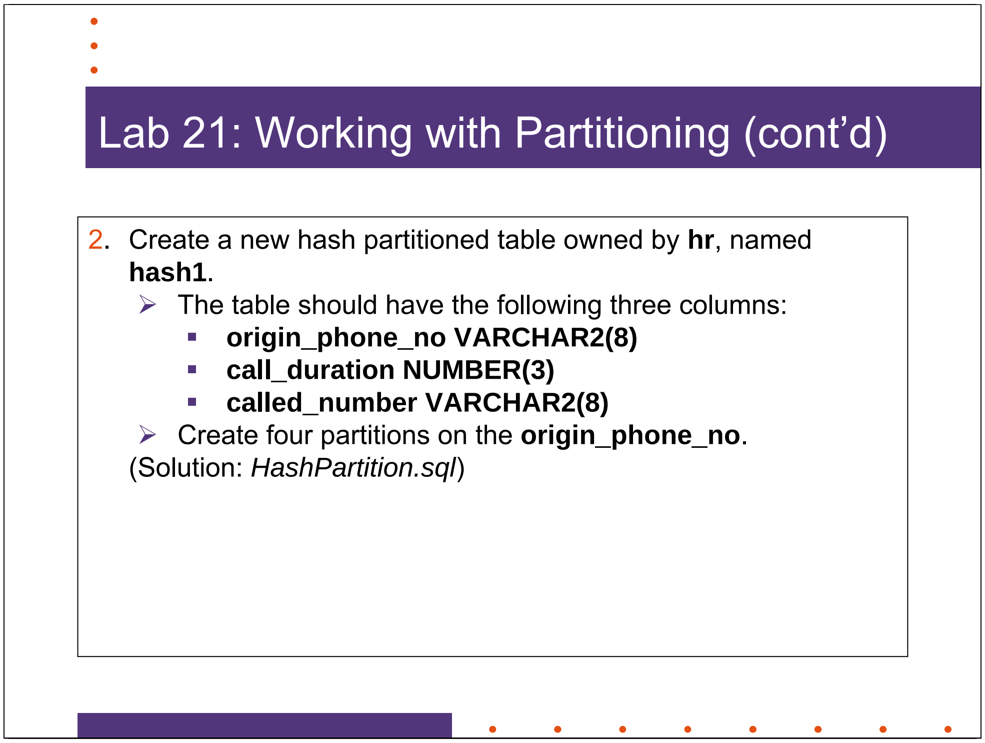
Sample code to illustrate how this works.



Lab 21: Working with Partitioning

1. Create a new range partitioned table owned by **hr**, named **range1**.
   - The table should contain three columns:
     - **department_id NUMBER(3)**
     - **first_name VARCHAR2(20)**
     - **last_name VARCHAR2(20)**
   - Specify at least four partitions on **department_id** with boundaries of **10**, **20**, etc.

   (Solution: *RangePartition.sql*)



Lab 21: Working with Partitioning (cont'd)

2. Create a new hash partitioned table owned by **hr**, named **hash1**.
   - The table should have the following three columns:
     - **origin_phone_no VARCHAR2(8)**
     - **call_duration NUMBER(3)**
     - **called_number VARCHAR2(8)**
   - Create four partitions on the **origin_phone_no**.

   (Solution: *HashPartition.sql*)