5

Managing Database Storage Structures

Terminal Learning Objective

ACTION: Maintain Oracle Database Storage Structures

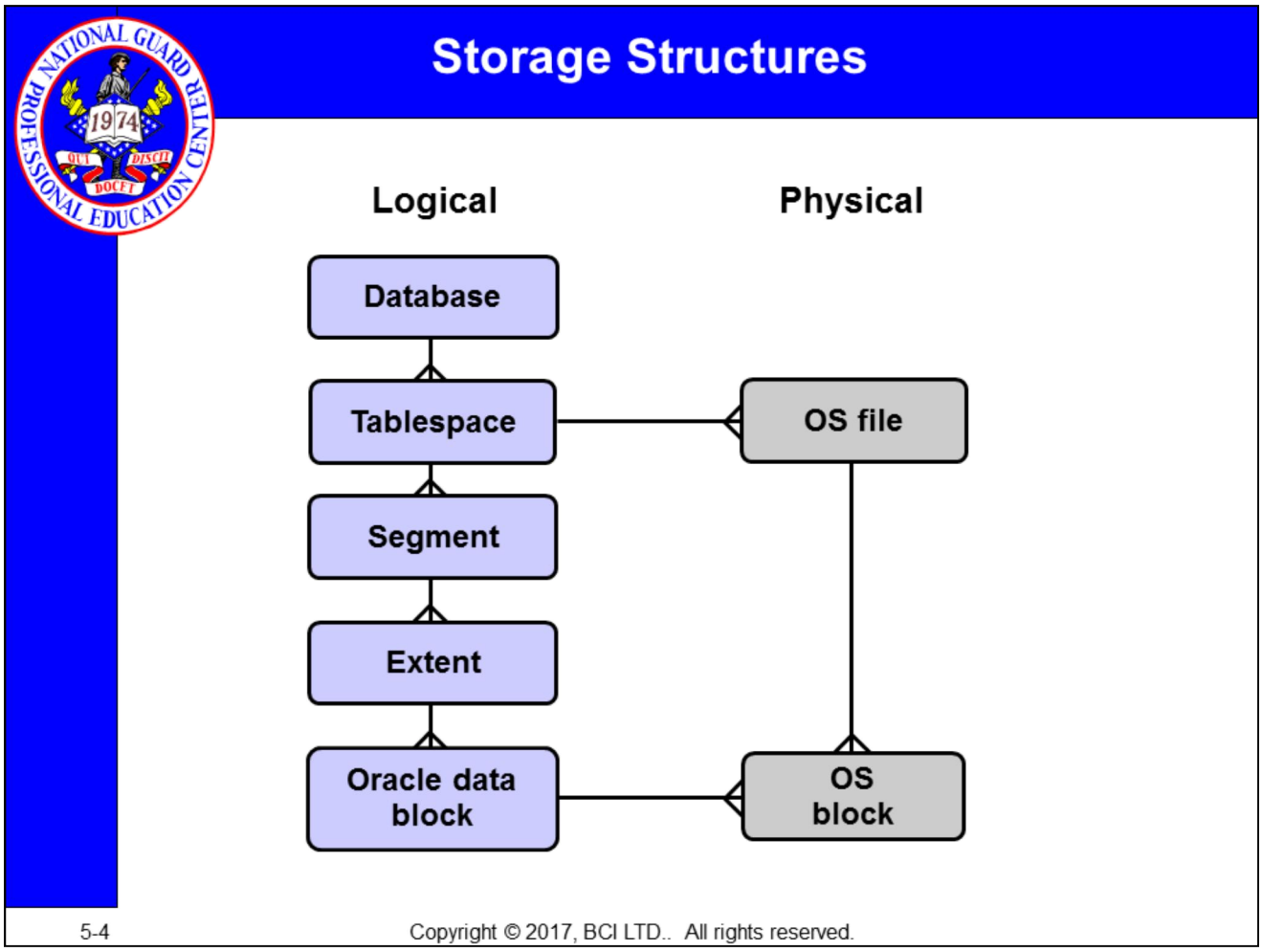
CONDITION: Given a student handout and Oracle DBA Handbook

STANDARD: Students will successfully create tablespaces, tables and indexes

Lesson Overview

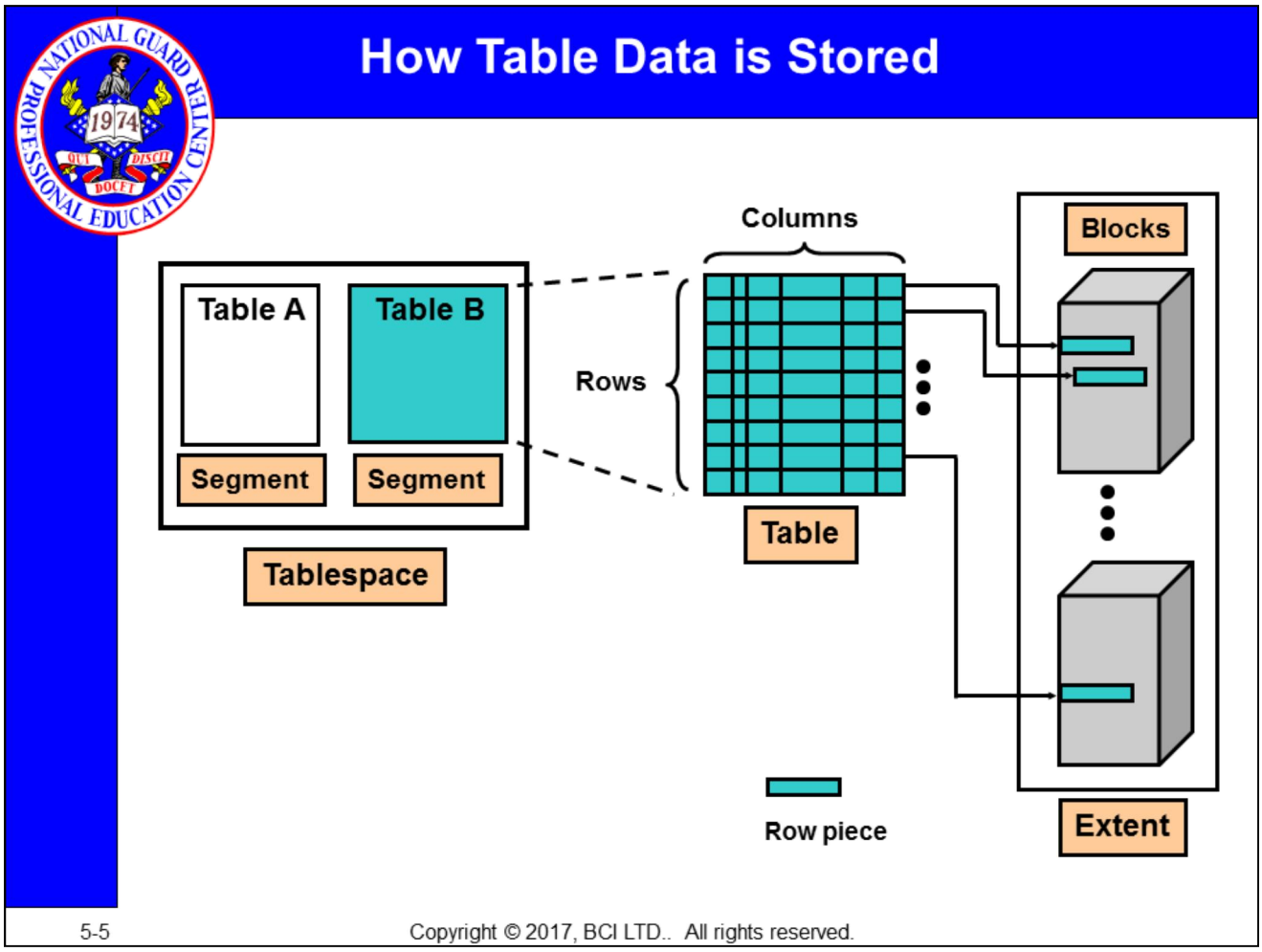
After completing this lesson, you should be able to:

- Describe how table row data is stored in blocks
- Define the purpose of tablespaces and data files
- Create and manage tablespaces
- Obtain tablespace information



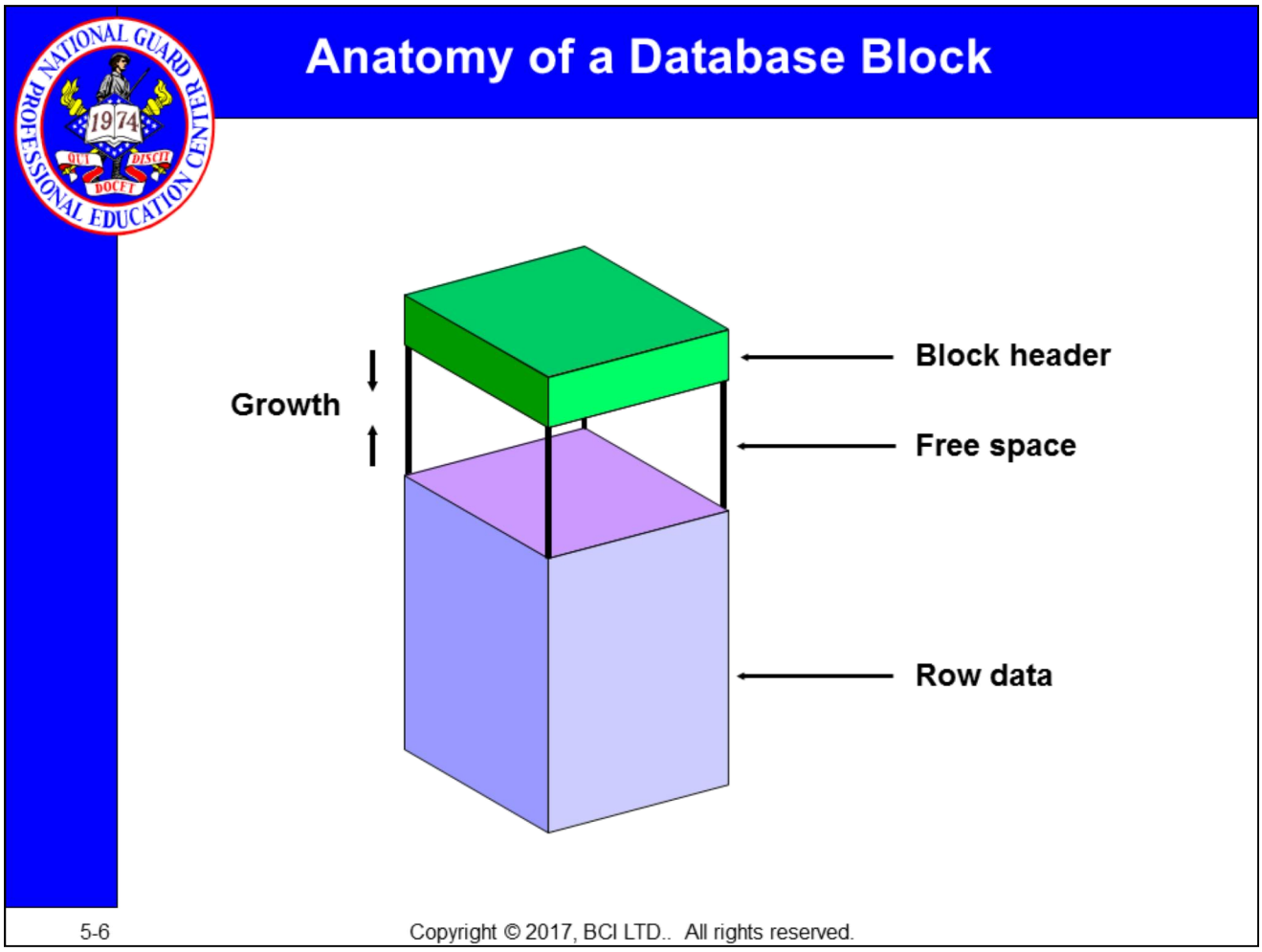
Storage Structures

A database is divided into logical storage units called tablespaces. Each tablespace has many logical Oracle data blocks. The `DB_BLOCK_SIZE` parameter specifies how large a logical block is. A logical block can range from 2 KB to 32 KB in size. The default size is 8 KB. A specific number of contiguous logical blocks form an extent. A set of extents that are allocated for a certain logical structure form one segment. An Oracle data block is the smallest unit of logical I/O.



How Table Data Is Stored

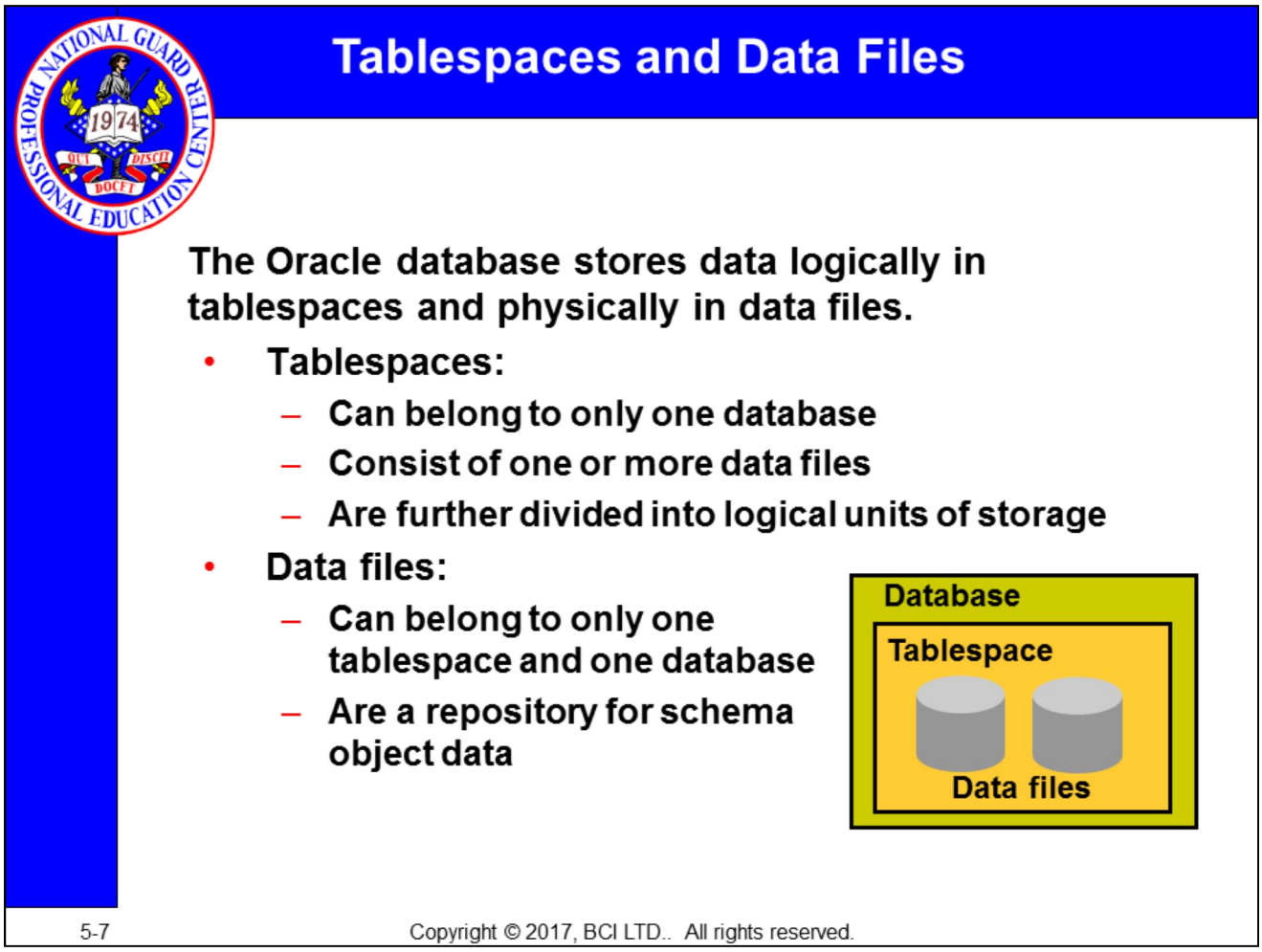
When a table is created, a segment is created to hold its data. A tablespace contains a collection of segments. Logically, a table contains rows of column values. A row is ultimately stored in a database block in the form of a row piece. It is called a row piece because under some circumstances the entire row may not be stored in one place. This happens when an inserted row is too large to fit into a single block or when an update causes an existing row to outgrow its current space.



Database Block: Contents

Oracle data blocks contain the following:

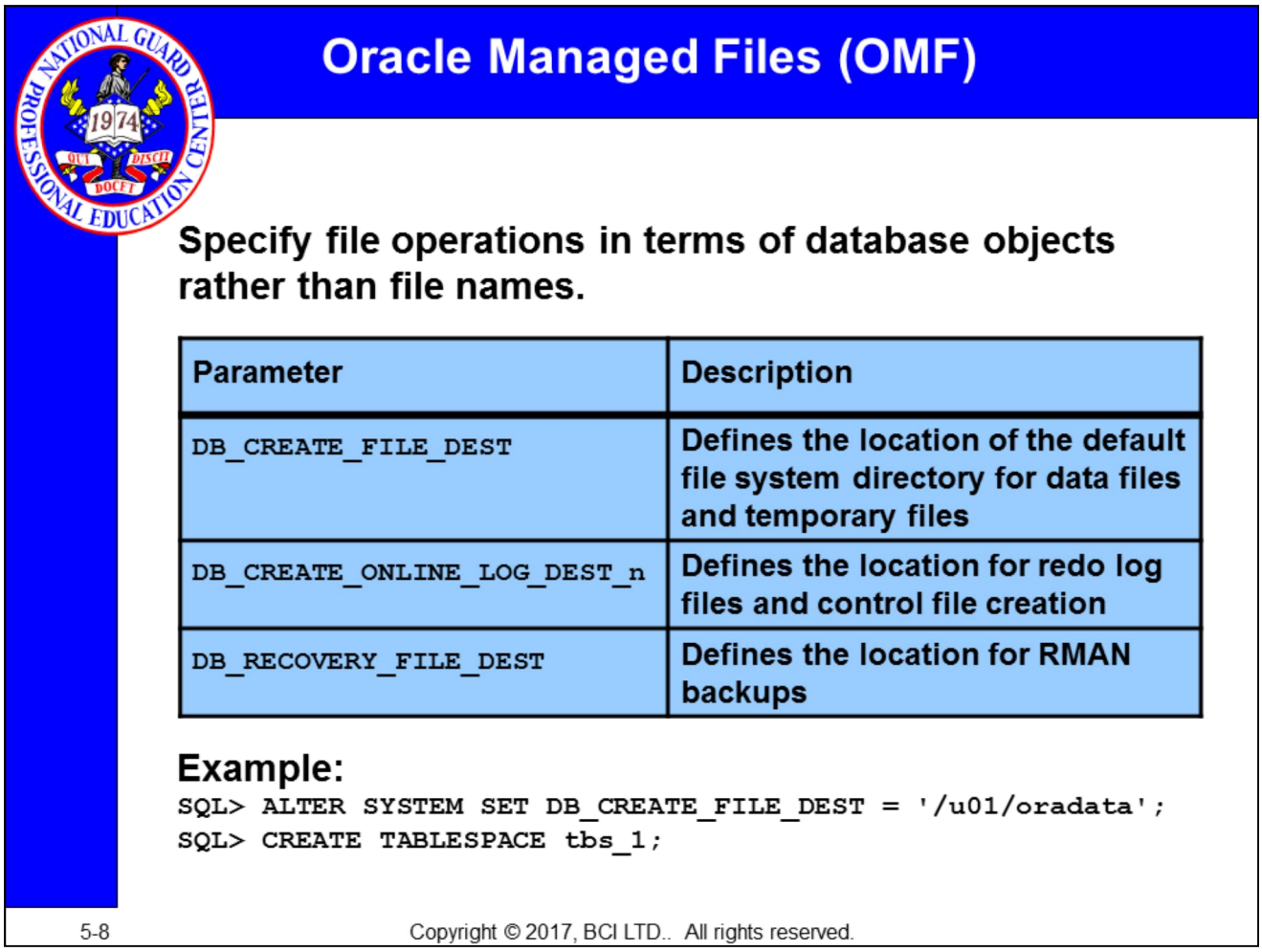
- **Block header:** The block header contains the segment type (such as table or index), data block address, table directory, row directory, and transaction slots of size 23 bytes each, which are used when modifications are made to rows in the block. The block header grows downward from the top.
- **Row data:** This is the actual data for the rows in the block. Row data space grows upward from the bottom.
- **Free space:** Free space is in the middle of the block. This enables the header and the row data space to grow when necessary. Row data takes up free space as new rows are inserted or columns of existing rows are updated with larger values. The examples of events that cause header growth are when the row directory needs more row entries or more transaction slots are required than initially configured. Initially, the free space in a block is contiguous. However, deletions and updates may fragment the free space in the block. The free space in the block is coalesced by the Oracle server when necessary.



Tablespaces and Data Files

Databases, tablespaces, and data files are closely related but they have important differences:

- An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all the database's data.
- Each tablespace in an Oracle database consists of one or more files called data files, which are physical structures that conform to the operating system on which the Oracle software runs.
- A database's data is collectively stored in the data files that constitute each tablespace of the database. For example, the simplest Oracle database would have two tablespaces (the required `SYSTEM` and `SYSAUX` tablespaces), each with one data file. Another database can have three tablespaces, each consisting of two data files (for a total of six data files). A single database can potentially have as many as 65,534 data files.



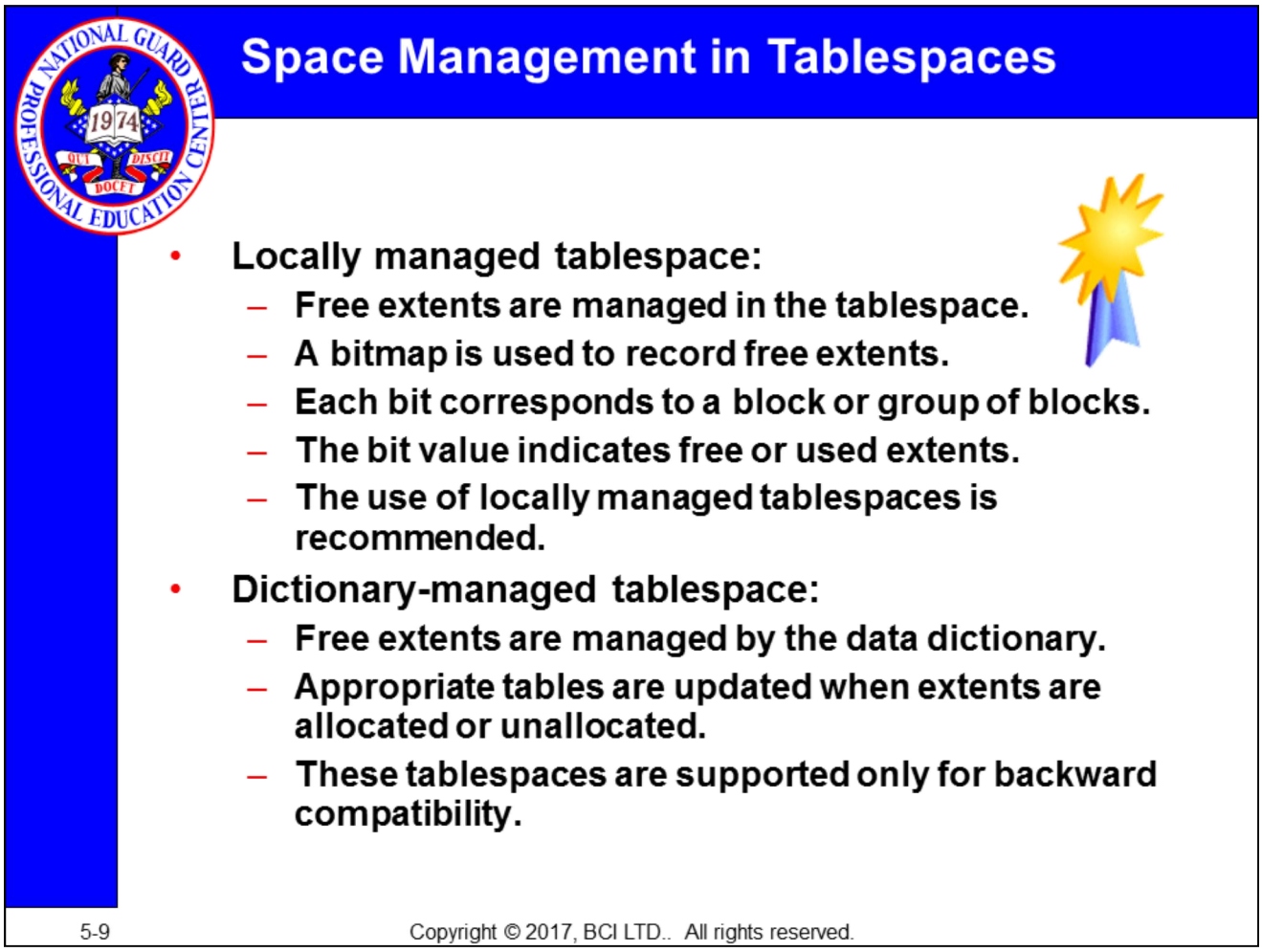
Oracle Managed Files (OMF)

Oracle Managed files (OMF) eliminate the need for you, to directly manage the operating system files comprising an Oracle database. You specify operations in terms of database objects rather than file names. The database internally uses standard file system interfaces to create and delete files as needed for the following database structures:

- Tablespaces
- Redo log files
- Control files
- Archived logs
- Block change tracking files
- Flashback logs
- RMAN backups

A database can have a mixture of Oracle-managed and unmanaged files. The file system directory specified by either of these parameters must already exist: the database does not create it. The directory must also have permissions to allow the database to create the files in it.

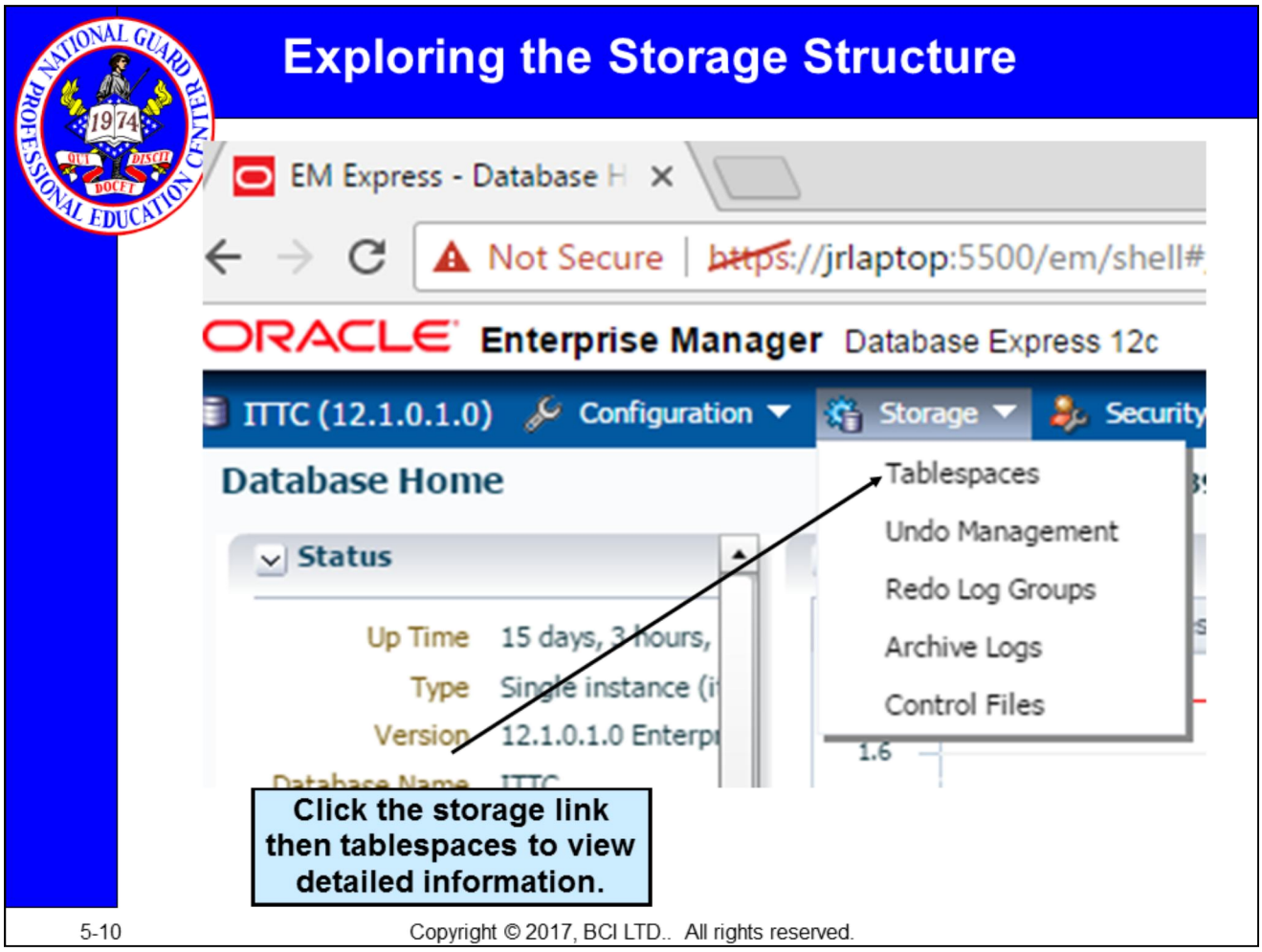
The example shows that after `DB_CREATE_FILE_DEST` is set, the `DATAFILE` clause can be omitted from a `CREATE TABLESPACE` statement. The data file is created in the location specified by `DB_CREATE_FILE_DEST`.



Space Management in Tablespaces

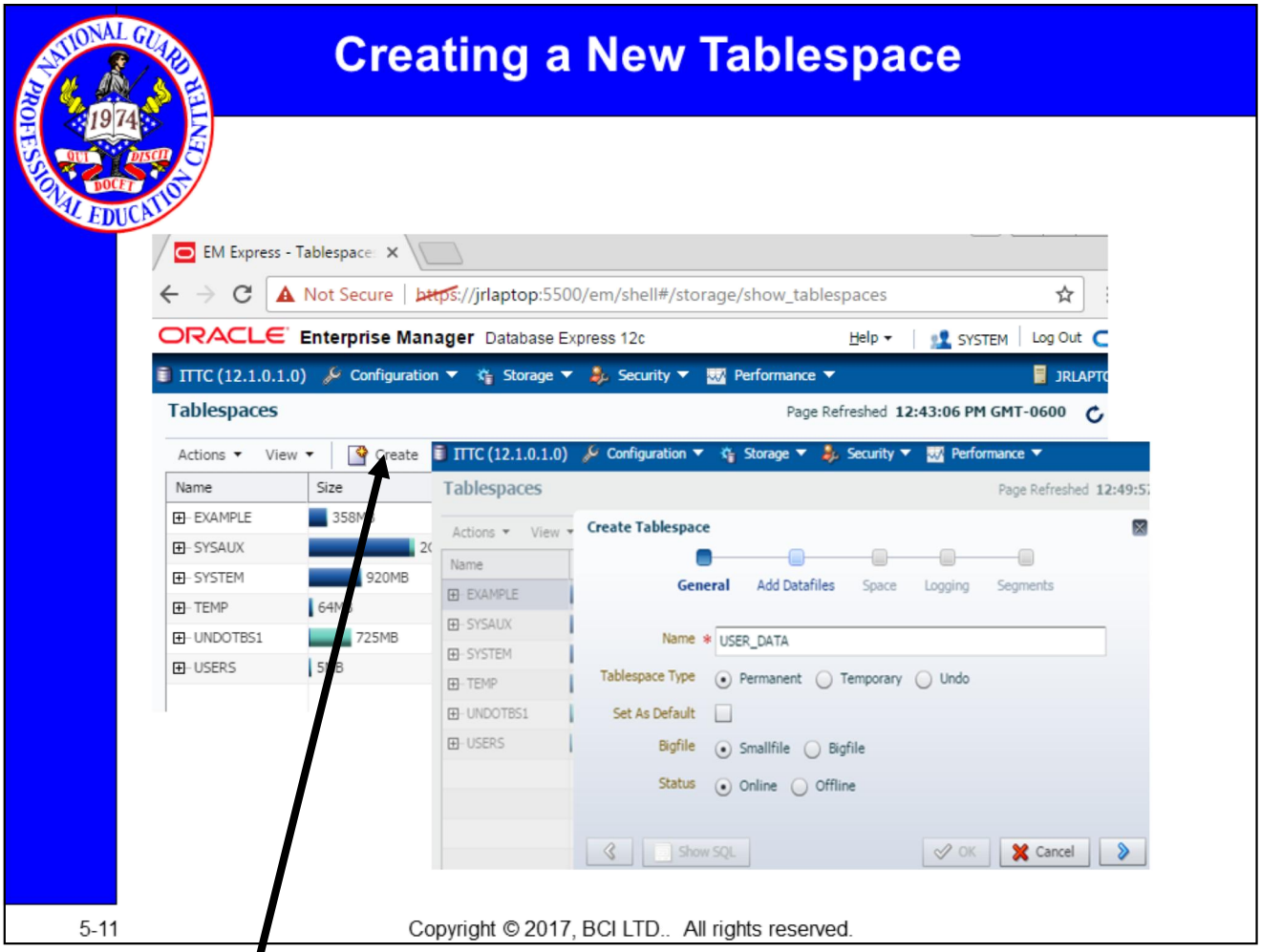
Tablespaces allocate space in extents. Tablespaces can be created to use one of the following two methods of keeping track of free and used space:

- **Locally managed tablespaces:** The extents are managed within the tablespace via bitmaps. Each bit in the bitmap corresponds to a block or a group of blocks. When an extent is allocated or freed for reuse, the Oracle server changes the bitmap values to show the new status of the blocks.
- **Dictionary-managed tablespaces:** The extents are managed by the data dictionary. The Oracle server updates the appropriate tables in the data dictionary whenever an extent is allocated or unallocated. This is for backward compatibility; it is recommended that you use locally managed tablespaces.



Exploring the Storage Structure

Logical data structures are stored in the physical files of the database. You can easily view the logical structures of your database through Enterprise Manager. Detailed information about each structure can be obtained by clicking the links in the Storage region of the Administration page.



Creating a New Tablespace

To create a tablespace, perform the following steps:

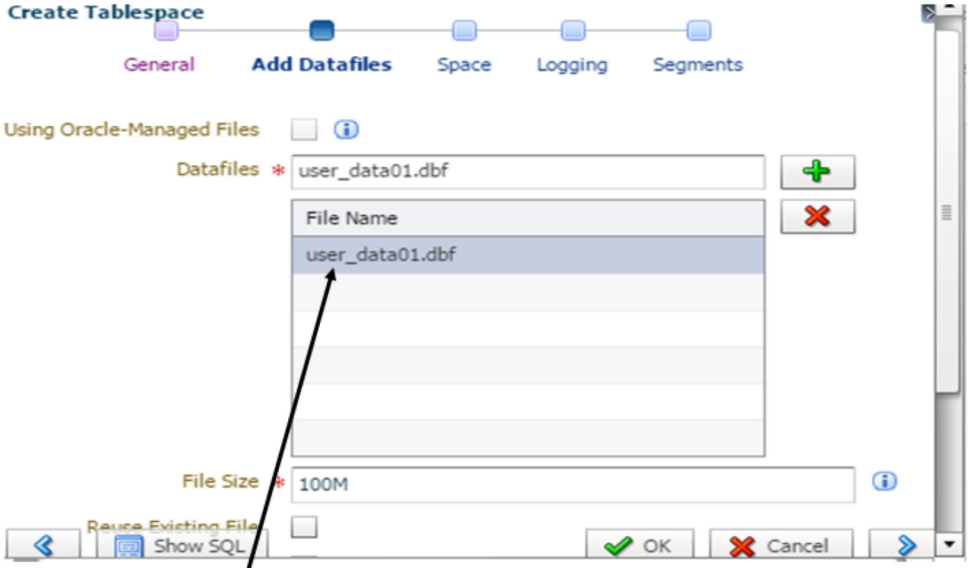
1. Click the Storage tab, and then click Tablespaces under the Storage heading.
2. Click Create. The Create Tablespace page appears.
3. Enter a name for the tablespace like USER_DATA.
4. Click on the button to go to the datafile naming page.
5. Under the datafile page, type user_data01.dbf. Note that Oracles naming convention adds a 01.dbf to the name of the tablespace.

Creating a New Tablespace (continued)

6. Under the Status heading, select Read Write. The Read Write status means that users can read and write to the tablespace after it is created. This is the default.
7. In the Datafiles region of the page, click Add to add data files to the tablespace; a tablespace must have at least one file. Bigfile tablespaces are used with extremely large databases, where Oracle's Automatic Storage Management (ASM) or other logical volume managers support striping or redundant array of independent disks (RAID) and dynamically extensible logical volumes.
8. On the Add Datafiles page, enter a file name. Accept the default for File Directory, and enter a file size.
9. In the Storage region, you can select "Automatically extend datafile when full (AUTOEXTEND)" and then specify an amount in the Increment field. This causes the data file to extend automatically each time it runs out of space. It is limited, of course, by the physical media on which it resides. Leave Maximum File Size as Unlimited. Click OK. You are returned to the Create Tablespace page.
10. Click the Storage tab. The Edit Tablespace page appears.
11. Accept all the defaults on the Storage page.

Note: These steps are intended to show you how to quickly create a tablespace for most situations. You may need to change some options, depending on your particular requirements.

Storage for Locally Managed Tablespaces



Click on the + button to place the new datafile name In the File Name line as shown. Then click on the › button to move to the space page.

Storage for Locally Managed Tablespaces (continued)

Advantages of Locally Managed Tablespaces

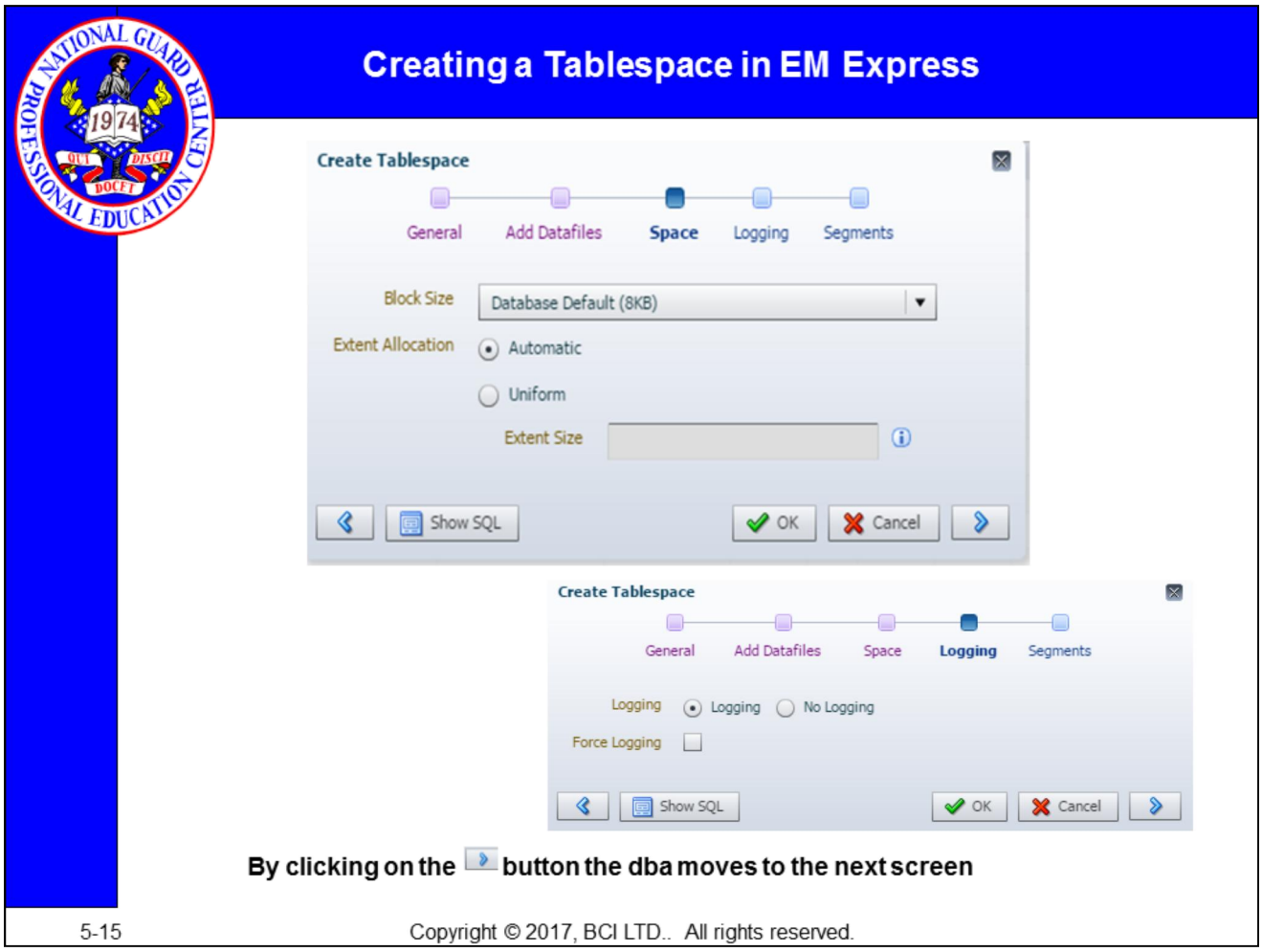
Locally managed tablespaces have the following advantages over dictionary-managed tablespaces:

- Local management avoids recursive space management operations. This occurs in dictionary-managed tablespaces if consuming or releasing space in an extent results in another operation that consumes or releases space in an undo segment or data dictionary table.
- Because locally managed tablespaces do not record free space in data dictionary tables, they reduce contention on these tables.
- Local management of extents automatically tracks adjacent free space, eliminating the need to coalesce free extents.
- The sizes of extents that are managed locally can be determined automatically by the system.
- Changes to the extent bitmaps do not generate undo information because they do not update tables in the data dictionary (except for special cases such as tablespace quota information).

Note: If you are managing a database that has dictionary-managed tablespaces and you want to convert them to locally managed tablespaces, use the `DBMS_SPACE_ADMIN.TABLESPACE_MIGRATE_TO_LOCAL` procedure to do this. For details about the use of this procedure, see *PL/SQL Packages and Types Reference* and the *Database Administrator's Guide*.

Block Information

This region shows the block size that is used for the tablespace being created. It is displayed here as a read-only value. If you set any of the alternate block size initialization parameters (`DB_nK_CACHE_SIZE`), then those other values would be listed here as an option. For more information about defining other block sizes, see the *Oracle Database Administrator's Guide*.



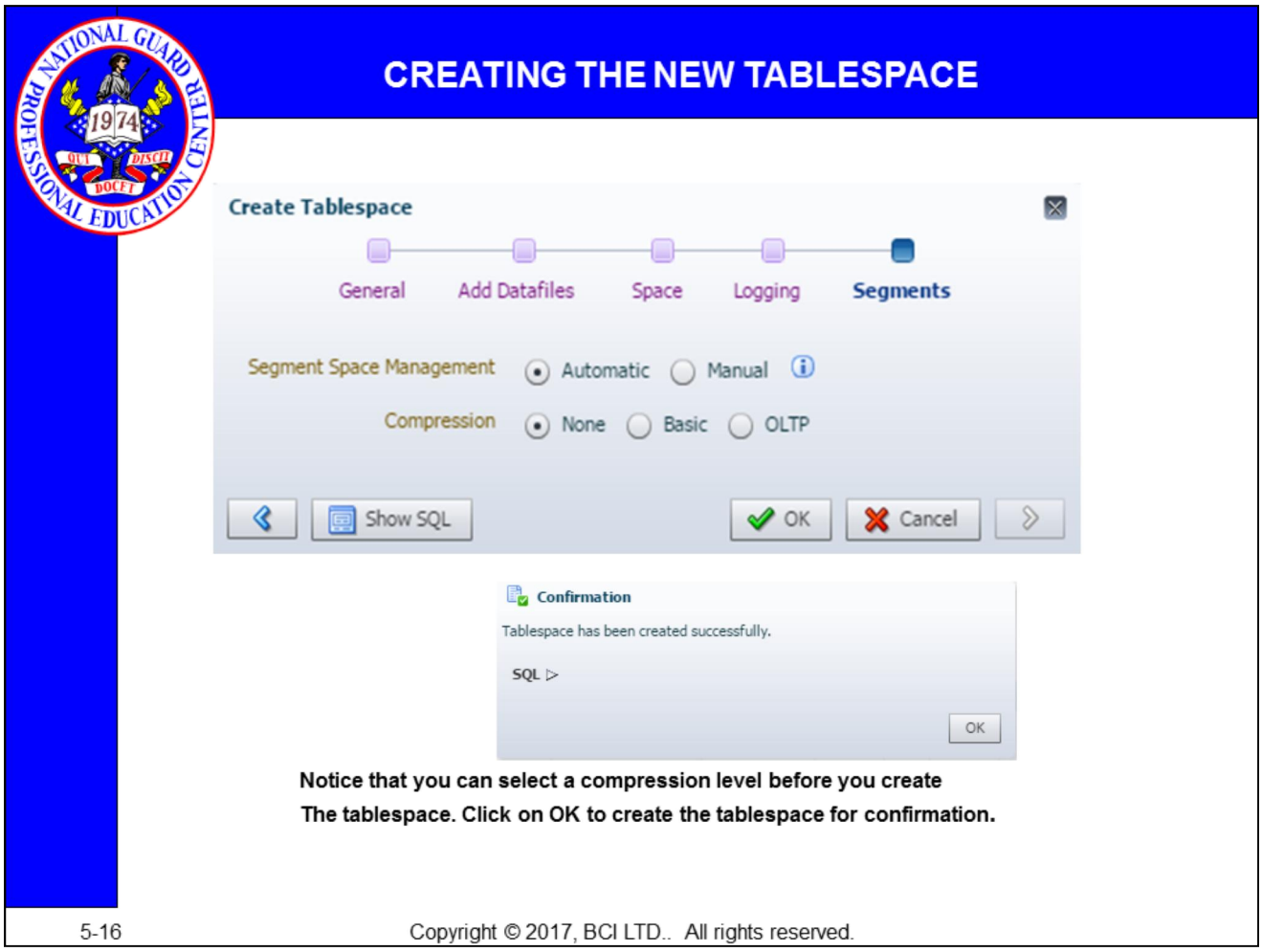
The DBA by clicking on the button moves to the next screen where it shows Whether or not the tablespace is to log all activity. The only time nologging is Used is for Index tablespaces where for performance reasons the DBA does Not want to log updates in order to speed up index creation and rebuilding.

Click the to move to the last page on segments.

NOTE: **Logging**

Changes made to objects in the tablespace are written to the redo log. If logging is not enabled, then the creation of any objects and any subsequent direct load path inserts performed on those objects are not written to the redo log, and the objects are thus unrecoverable in the event of data loss. So, when an object is created without Logging enabled, you must back up those objects, if you want them to be recoverable.

For more details about the logging clause, see the *Oracle Database SQL Reference*.



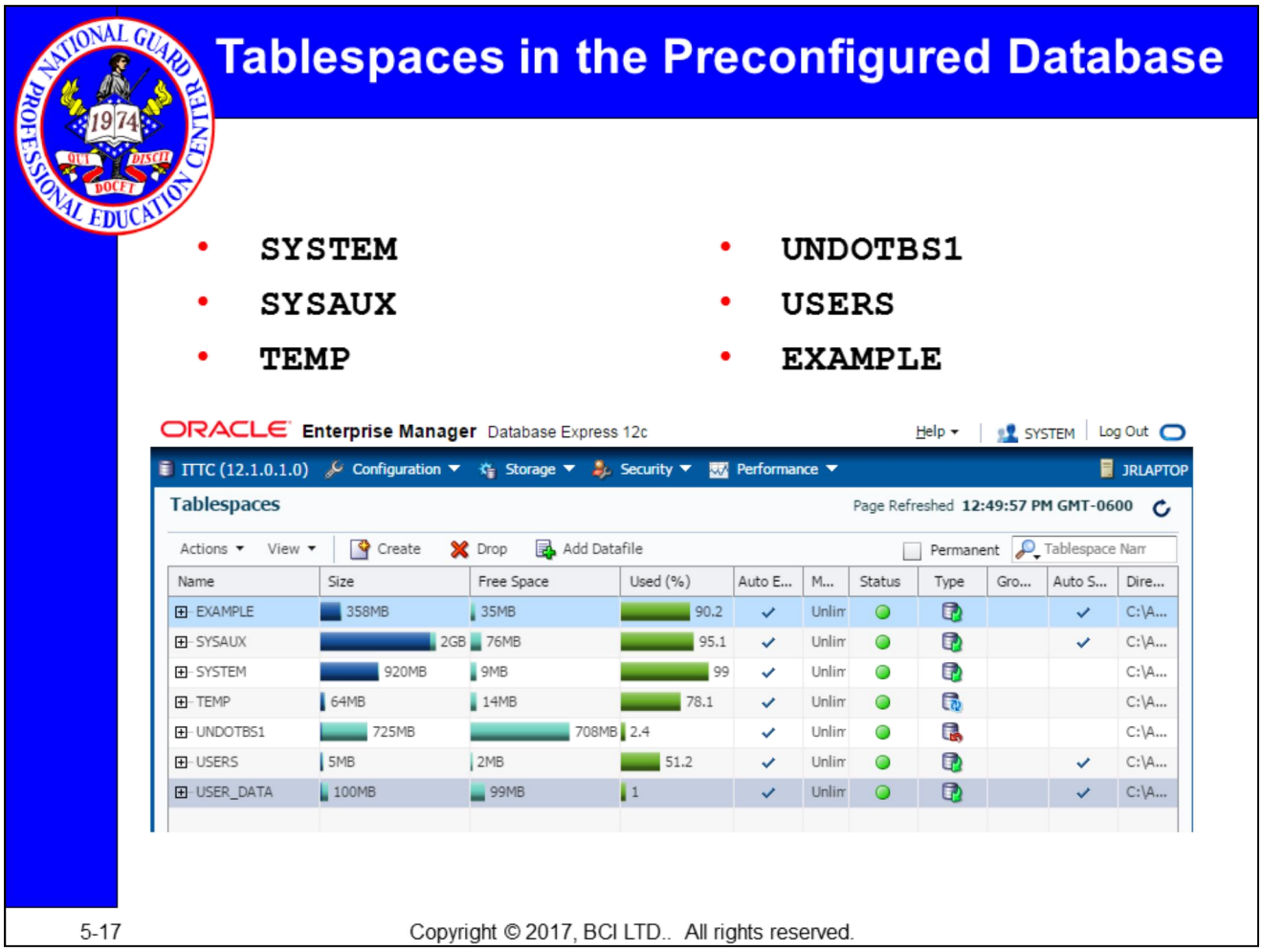
Storage for Locally Managed Tablespaces

The extents within a locally managed tablespace can be allocated in one of these two ways:

- **Automatic:** Also called auto allocate, it specifies that the sizes of the extents within the tablespace are system managed. You cannot specify an extent size. You cannot specify Automatic for a temporary tablespace.
- **Uniform:** It specifies that the tablespace is managed with uniform extents of a size that you specify. The default size is 1 MB. All extents of temporary tablespaces are uniform and default to that value. You cannot specify Uniform for an undo tablespace.

Segment space management within a locally managed tablespace can be specified as:

- **Automatic:** The Oracle database uses bitmaps to manage the free space within segments. The bitmap describes the status of each data block within a segment with respect to the amount of space in the block that is available for inserting rows. As more or less space becomes available in a data block, its new state is reflected in the bitmap. With bitmaps, the Oracle database manages free space more automatically and, thus, this form of space management is called Automatic Segment Space Management (ASSM).



Tablespaces in the Preconfigured Database

The following tablespaces are created in the preconfigured database in this course:

- **SYSTEM:** The SYSTEM tablespace is used by the Oracle server to manage the database. It contains the data dictionary and tables that contain administrative information about the database. These are all contained in the SYS schema and can be accessed only by the SYS user or other administrative users with the required privilege.
- **SYSAUX:** This is an auxiliary tablespace to the SYSTEM tablespace. Some components and products that used the SYSTEM tablespace or their own tablespaces in earlier releases of the Oracle database, now use the SYSAUX tablespace. Every Oracle Database12c or later release must have a SYSAUX tablespace.

  In Enterprise Manager, you can see a pie chart of the contents of this tablespace. To do this, click Tablespaces on the Administration page. Select SYSAUX and click Edit. Then, click the Occupants tab. After creation, you can monitor the space usage of each occupant inside the SYSAUX tablespace by using EM. If you detect that a component is taking too much space in the SYSAUX tablespace, or if you anticipate that it will, you can move the occupant into a different tablespace by selecting one of the occupants and clicking Change Tablespace.

Tablespaces in the Preconfigured Database (continued)

- **TEMP:** Your temporary tablespace is used when you execute a SQL statement that requires the creation of temporary segments (such as a large sort or the creation of an index). Just like each user is assigned a default tablespace for storing created data objects, each user is assigned a temporary tablespace. The best practice is to define a default temporary tablespace for the database, which is assigned to any newly created user, unless otherwise specified. In the preconfigured database, the `TEMP` tablespace is specified as the default temporary tablespace. This means that if no temporary tablespace is specified when the user account is created, the Oracle database assigns this tablespace to the user.
- **UNDOTBS1:** This is the undo tablespace used by the database server to store undo information. If a database uses Automatic Undo Management, then it must have exactly one active undo tablespace at any given time. This tablespace is created at database creation time.
- **USERS:** This tablespace is used to store permanent user objects and data. In the preconfigured database, the `USERS` tablespace is the default tablespace for all objects created by nonsystem users. For the `SYS` and `SYSTEM` users (the system users), the default permanent tablespace remains `SYSTEM`.
- **EXAMPLE:** This tablespace contains the sample schemas that can be installed when you create the database. The sample schemas provide a common platform for examples. Oracle documentation and courseware contain examples based on the sample schemas.

Note: To simplify administration, it is common to have a tablespace for indexes alone.

Altering a Tablespace

After you create a tablespace, you can later alter it in several ways as the needs of your system change.

Renaming: Enter a new name for the tablespace and click Apply.

Changing the status: A tablespace can be in one of three different statuses or states. Any of the following three states may not be available because their availability depends on the type of tablespace:

- **Read Write:** The tablespace is online and can be read from and written to.
- **Read Only:** Specify read-only to place the tablespace in transition read-only mode. In this state, existing transactions can be completed (committed or rolled back), but no further data manipulation language (DML) operations are allowed on objects in the tablespace. The tablespace is online while in the read-only state. You cannot make the `SYSTEM` or `SYSAUX` tablespace read-only.

Altering a Tablespace (continued)

- **Offline:** You can take an online tablespace offline so that this portion of the database is temporarily unavailable for general use. The rest of the database is open and available for users to access data. When you take it offline, you can use the following options:
  - **Normal:** A tablespace can be taken offline normally if no error conditions exist for any of the data files of the tablespace. The Oracle database ensures that all data is written to disk by taking a checkpoint for all data files of the tablespace as it takes them offline.
  - **Temporary:** A tablespace can be taken offline temporarily even if there are error conditions for one or more files of the tablespace. The Oracle database takes the data files (which are not already offline) offline, performing checkpointing on them as it does so. If no files are offline, but you use the temporary clause, media recovery is not required to bring the tablespace back online. However, if one or more files of the tablespace are offline because of write errors, and you take the tablespace offline temporarily, the tablespace requires recovery before you can bring it back online.
  - **Immediate:** A tablespace can be taken offline immediately without the Oracle database taking a checkpoint on any of the data files. When you specify Immediate, media recovery for the tablespace is required before the tablespace can be brought online. You cannot take a tablespace offline immediately if the database is running in `NOARCHIVELOG` mode.
  - **For Recover:** The `FOR RECOVER` setting has been deprecated. The syntax is supported for backward compatibility.

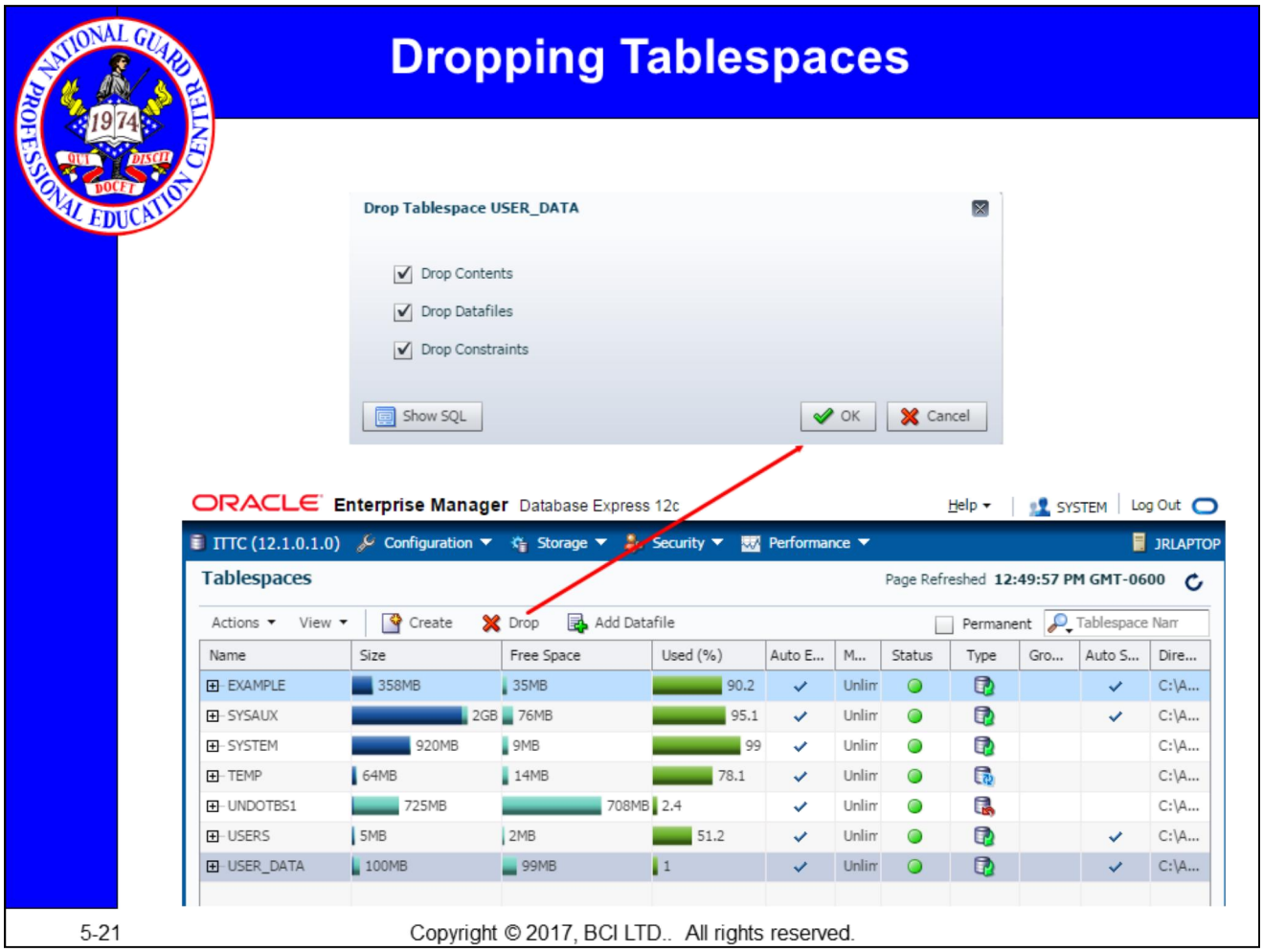
Changing the size: You can add space to an existing tablespace by either adding data files to the tablespace or by changing the size of an existing data file.

- To add a new data file to the tablespace, click Add and fill in the information about the data file on the Add Datafile page.
- To change the size of an existing data file, select the data file in the Datafiles region of the Edit Tablespace page by clicking the name of the data file, or select the data file and click Edit. Then, on the Edit Datafile page, you can change the size of the data file. You can make the tablespace either larger or smaller. However, you cannot make a data file smaller than the used space in the file; if you try to do so, you get the following error:

```
ORA-03297: file contains used data beyond requested RESIZE
value
```

Storage options: Click Storage to change the logging behavior of the tablespace.

- **Place Online:** Brings a currently offline tablespace online
- tablespace, including a graphical map of all of the extents
- **Take Offline:** Makes a currently online tablespace unavailable. The tablespace is not deleted or dropped; it is just unavailable.

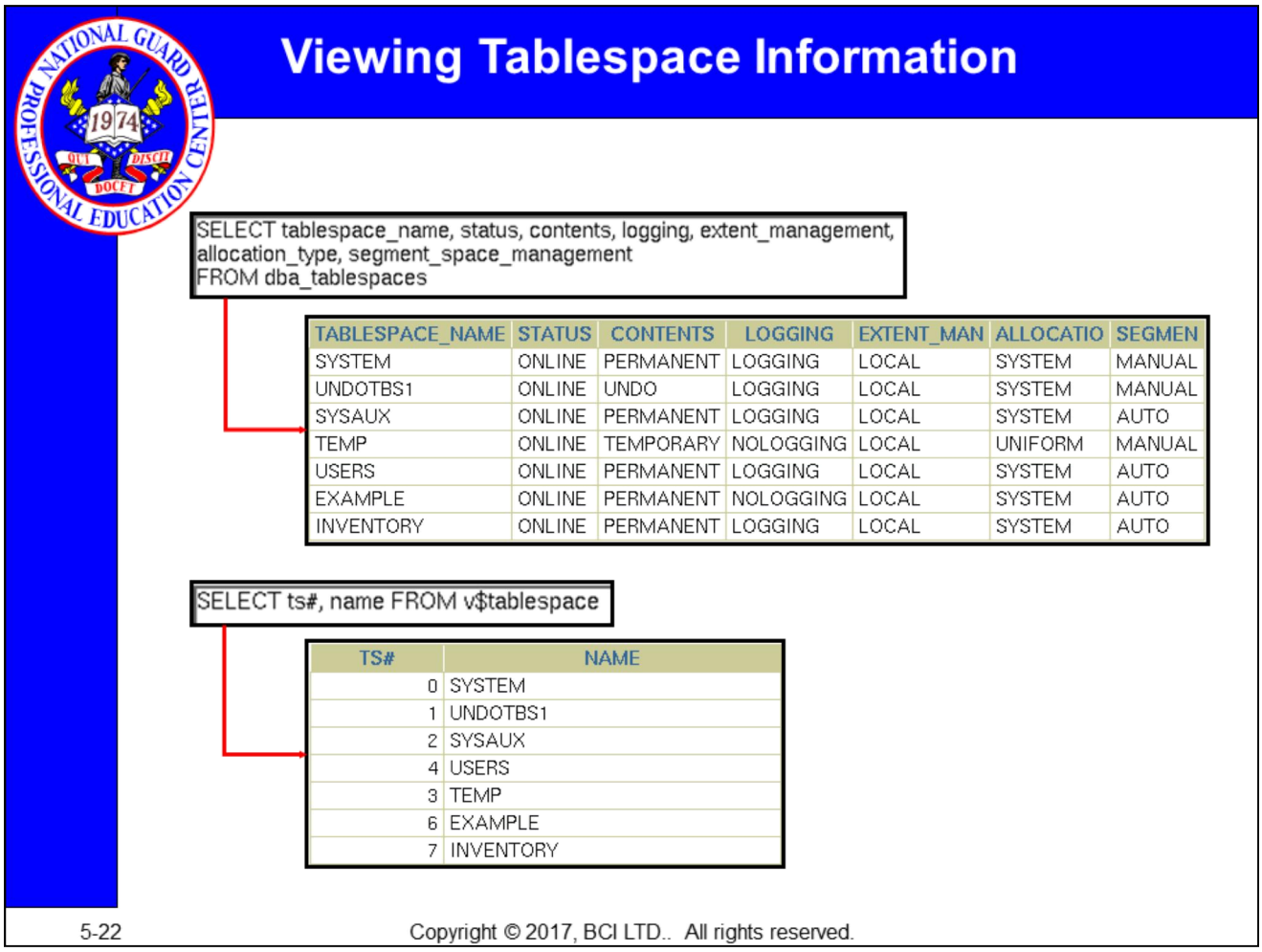


Dropping Tablespaces

You can drop a tablespace and its contents (the segments contained in the tablespace) from the database if the tablespace and its contents are no longer required. You must have the `DROP TABLESPACE` system privilege to drop a tablespace.

When you drop a tablespace, the file pointers in the control file of the associated database are removed. You can optionally direct the Oracle server to delete the operating system files (data files) that constitute the dropped tablespace. If you do not direct the Oracle server to delete the data files at the same time as it deletes the tablespace, you must later use the appropriate commands of your operating system if you want them to be deleted.

You cannot drop a tablespace that contains any active segments. For example, if a table in the tablespace is currently being used or the tablespace contains undo data that is needed to roll back uncommitted transactions, you cannot drop the tablespace. The tablespace can be online or offline, but it is best to take the tablespace offline before dropping it.

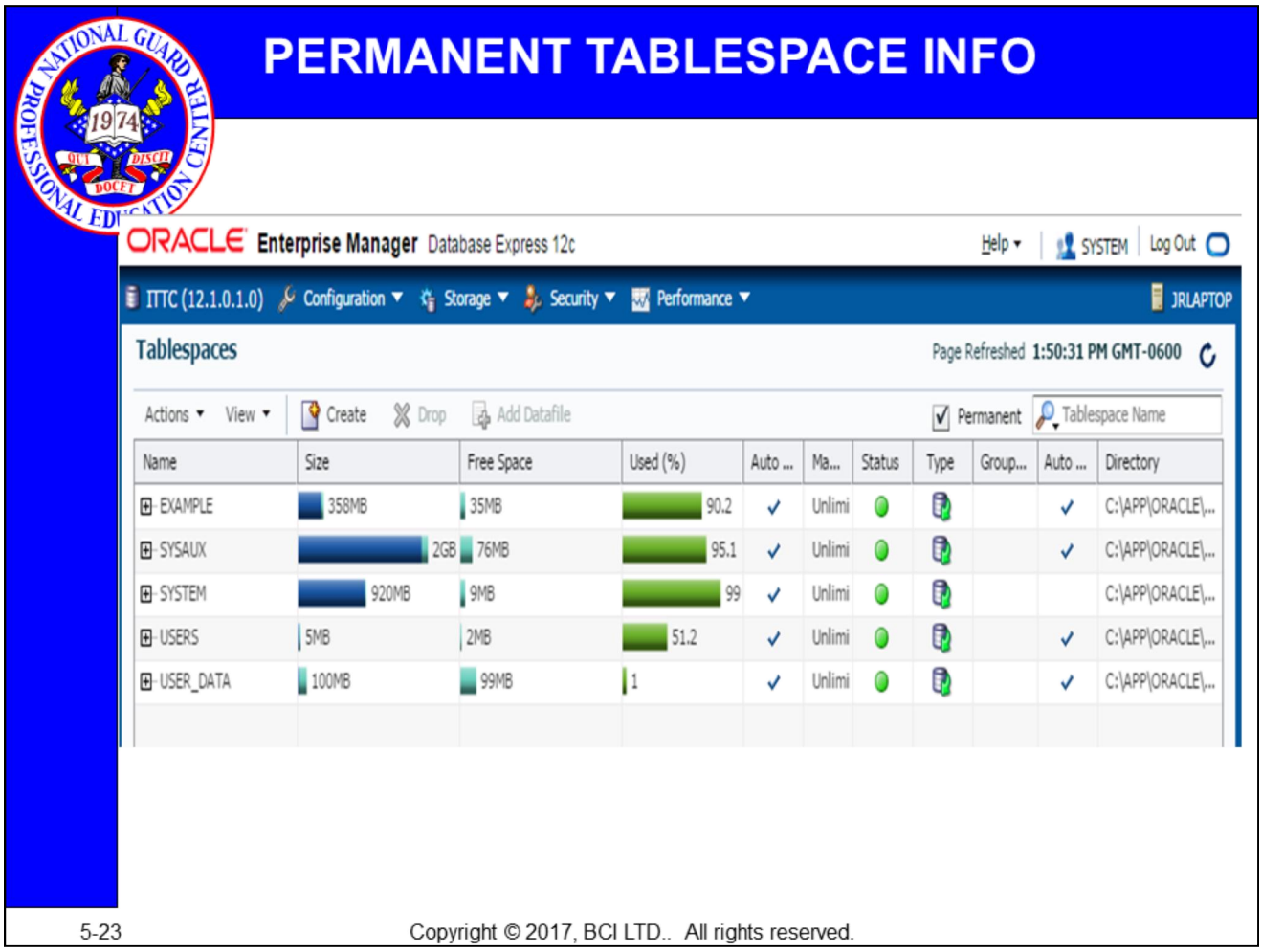


Viewing Tablespace Information

Click View to see information about the selected tablespace. On the View Tablespace page, you can also click Edit to alter the tablespace.

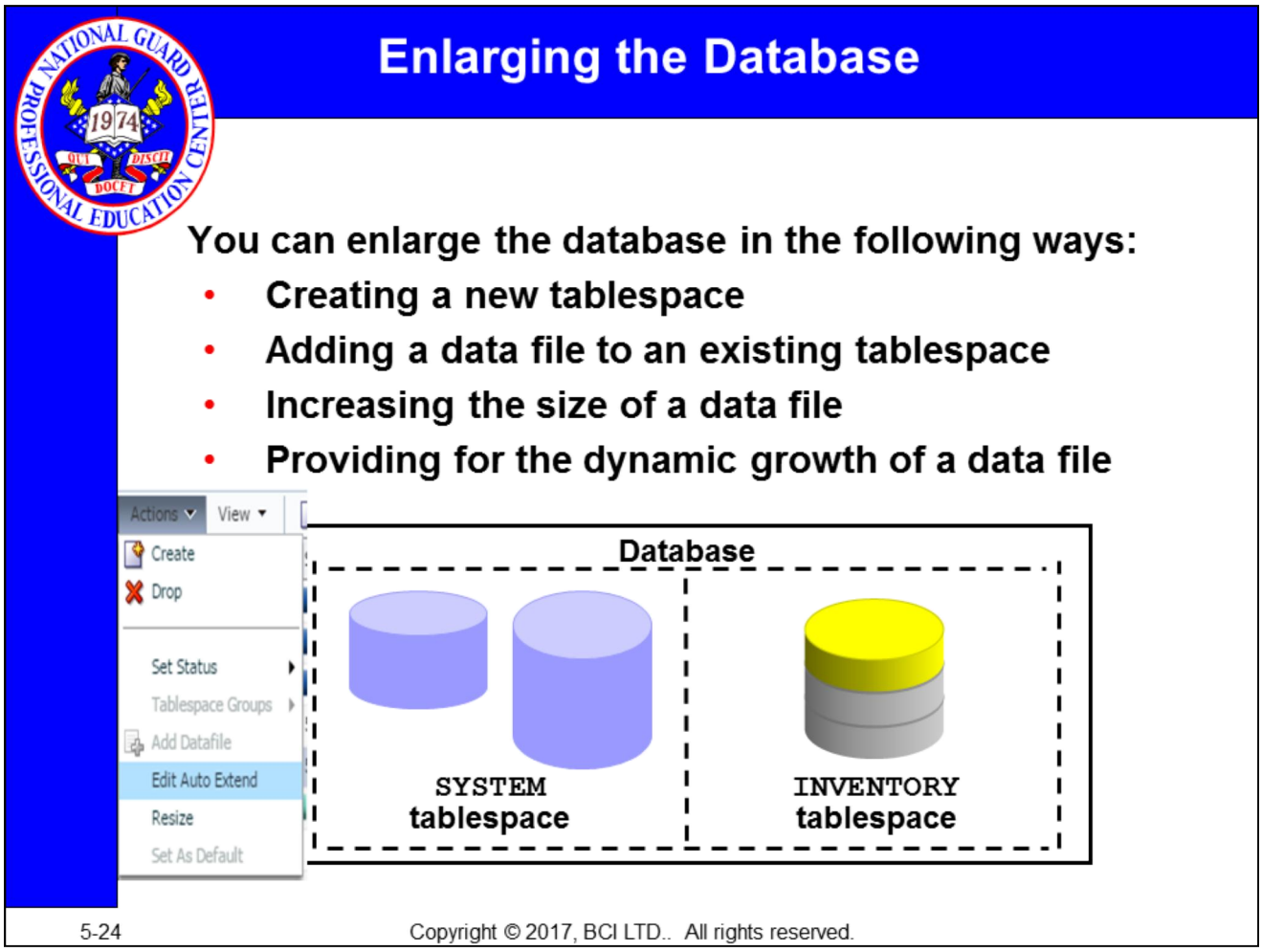
Tablespace and data file information can also be obtained by querying the following:

- **Tablespace information:**
  - `DBA_TABLESPACES`
  - `V$TABLESPACE`
- **Data file information:**
  - `DBA_DATA_FILES`
  - `V$DATAFILE`
- **Temp file information:**
  - `DBA_TEMP_FILES`
  - `V$TEMPFILE`



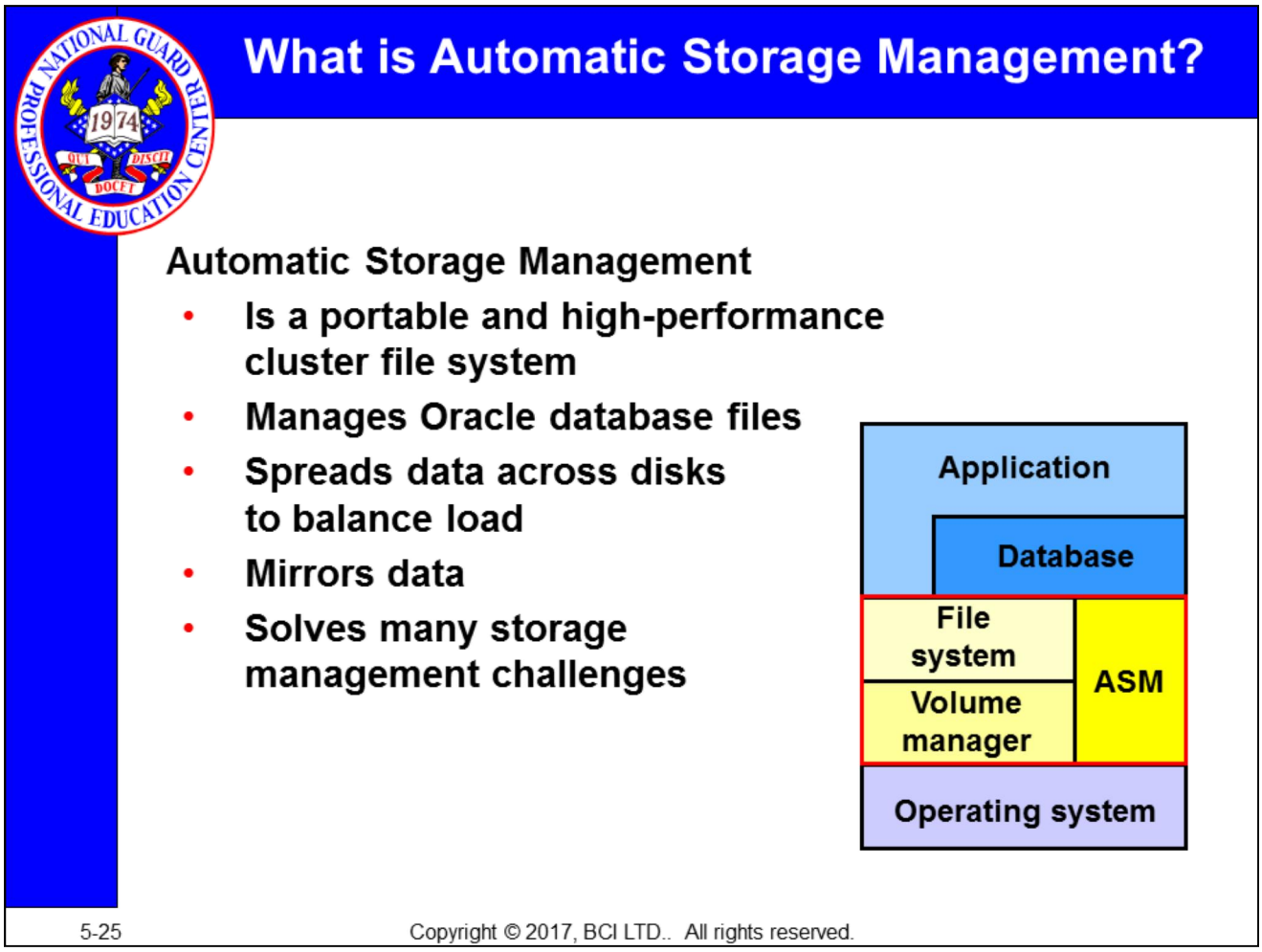
Gathering Storage Information

To view and modify just permanent tablespace information in EM Express, select the permanent box at top right. Then you can select any tablespace you wish.



Enlarging the Database

These activities can be performed with Enterprise Manager or with SQL statements. In the end, the size of the database can be described as the sum of all of its tablespaces.



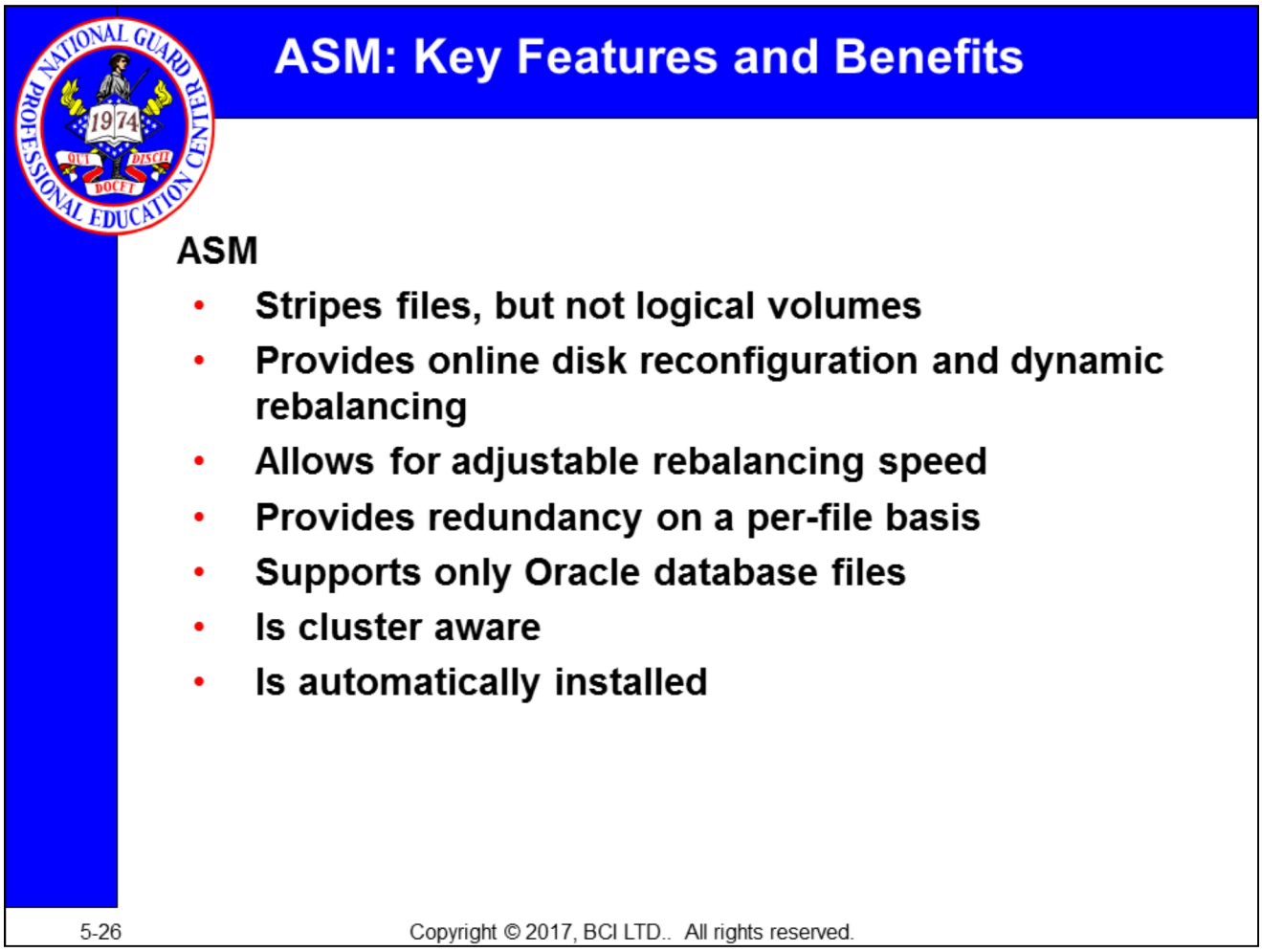
What Is Automatic Storage Management?

ASM provides a vertical integration of the file system and the volume manager that is specifically built for Oracle database files. ASM can provide management for single symmetric multiprocessing (SMP) machines or across multiple nodes of a cluster for Oracle Real Application Clusters (RAC) support.

ASM distributes input/output (I/O) load across all available resources to optimize performance while removing the need for manual I/O tuning. ASM helps DBAs manage a dynamic database environment by enabling them to increase the database size without having to shut down the database to adjust storage allocation.

ASM can maintain redundant copies of data to provide fault tolerance, or it can be built on top of vendor-supplied storage mechanisms. Data management is done by selecting the desired reliability and performance characteristics for classes of data rather than with human interaction on a per-file basis.

ASM capabilities save the DBA's time by automating manual storage and thereby increasing the DBA's ability to manage more and larger databases with increased efficiency.

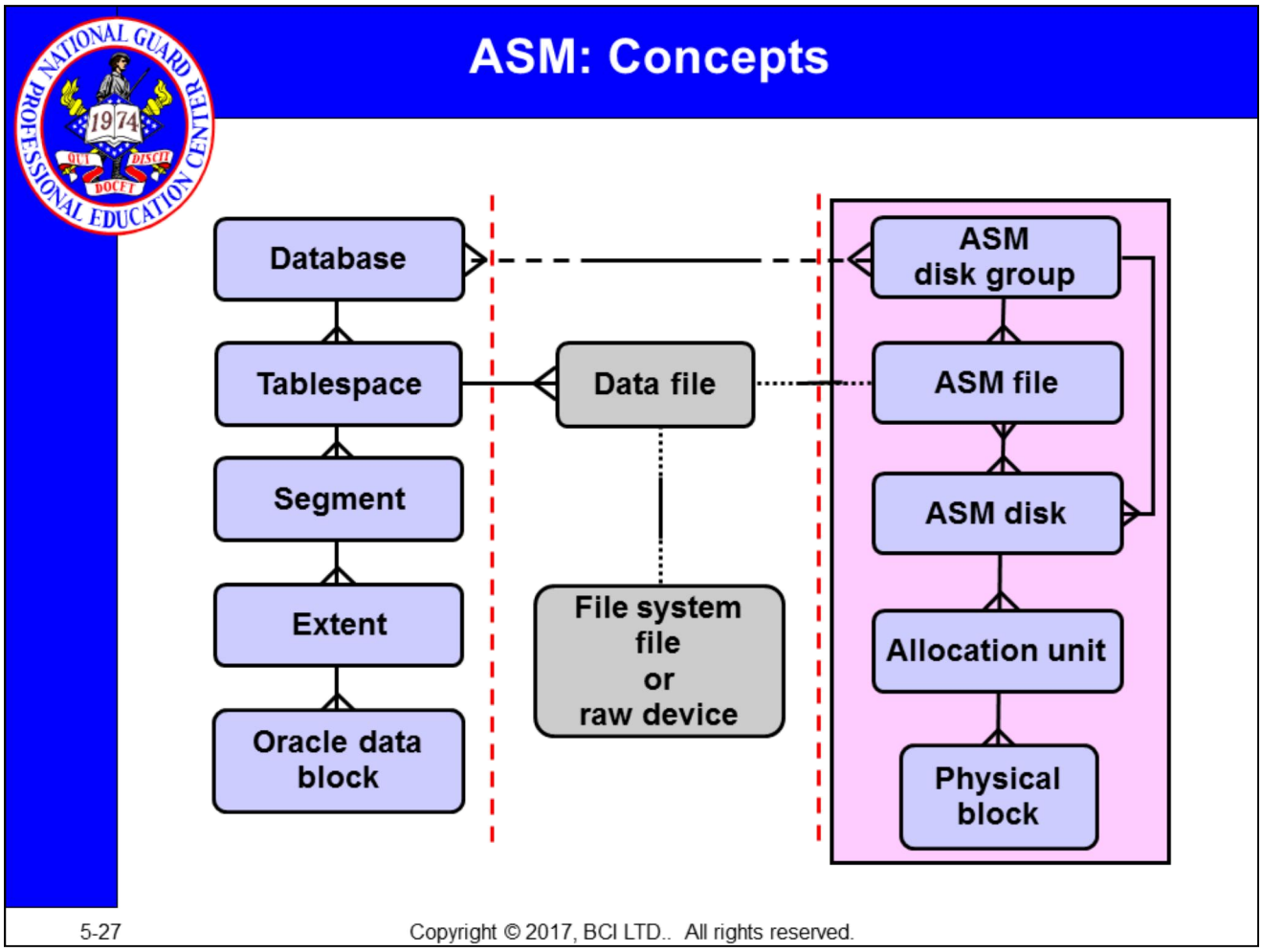


ASM: Key Features and Benefits

ASM divides files into extents (different from the data file extents discussed earlier) and spreads the extents for each file evenly across all the disks. It uses an index technique to track the placement of each extent. When the storage capacity changes, ASM does not restripe all the data but moves an amount of data proportional to the amount of storage added or removed to evenly redistribute the files and maintain a balanced load across the disks. This is done while the database is active.

You can increase the speed of a rebalance operation to cause it to finish sooner, or decrease the speed to reduce the impact on the I/O subsystem. ASM provides mirroring protection without the need to purchase a third-party Logical Volume Manager. One unique advantage of ASM is that mirroring is applied on a file basis, rather than on a volume basis. Therefore, the same disk group can contain a combination of mirrored or nonmirrored files.

ASM supports data files, log files, control files, archive logs, Recovery Manager (RMAN) backup sets, and other Oracle database file types. It supports RAC and eliminates the need for a Cluster Logical Volume Manager or a Cluster File System.



ASM: Concepts

ASM does not eliminate any preexisting database functionality. Existing databases are able to operate as they always have. You can create new files as ASM files and leave existing files to be administered in the old way, or you can eventually migrate them to ASM.

The diagram depicts the relationships that exist between the various storage components inside an Oracle database that uses ASM. The left and middle parts of the diagram show the relationships that exist in previous releases. On the right are the new concepts introduced by ASM.

Database files can be stored as ASM files. At the top of the new hierarchy are ASM disk groups. Any single ASM file is contained in only one disk group. However, a disk group may contain files belonging to several databases, and a single database may use storage from multiple disk groups. As you can see, one disk group is made up of multiple ASM disks, and each ASM disk belongs to only one disk group. ASM files are always spread across all the ASM disks in the disk group. ASM disks are partitioned in allocation units (AU) of one megabyte each. An allocation unit is the smallest contiguous disk space that ASM allocates. ASM does not allow an Oracle block to be split across allocation units.

Note: This graphic deals with only one type of ASM file: data file. However, ASM can be used to store other database file types.

Practical Exercise

This practical exercise covers the following topics:

- Creating tablespaces
- Gathering information about tablespaces

Summary

In this lesson, you should have learned how to:

- Describe how table row data is stored in blocks
- Define the purpose of tablespaces and data files
- Create and manage tablespaces
- Obtain tablespace information
- Describe the main concepts and functionality of Automatic Storage Management (ASM)

Terminal Learning Objective

ACTION: Maintain Oracle Database Storage Structures

CONDITION: Given a student handout and Oracle DBA Handbook

STANDARD: Students will successfully create tablespaces, tables and indexes