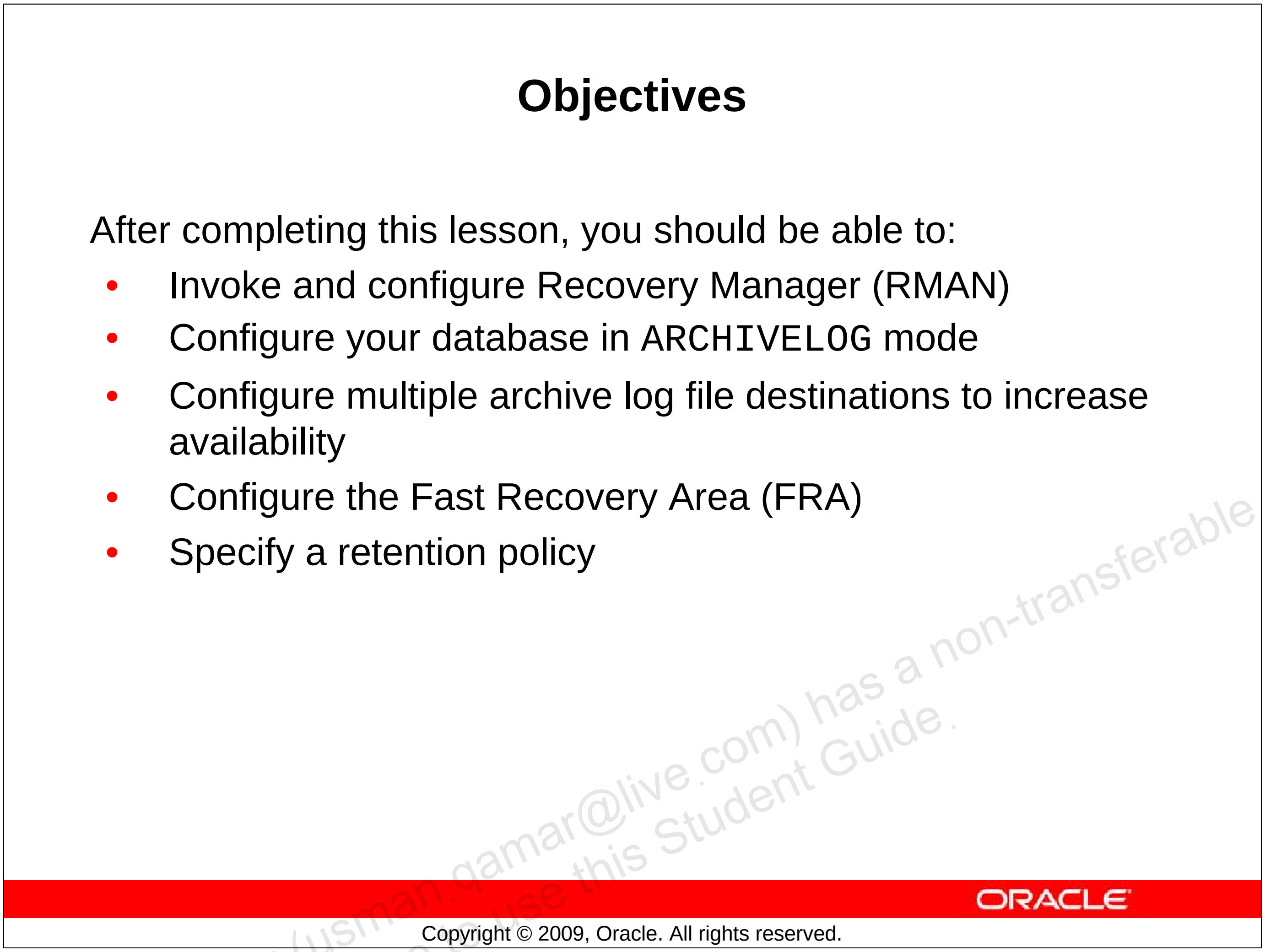
2

Configuring for Recoverability



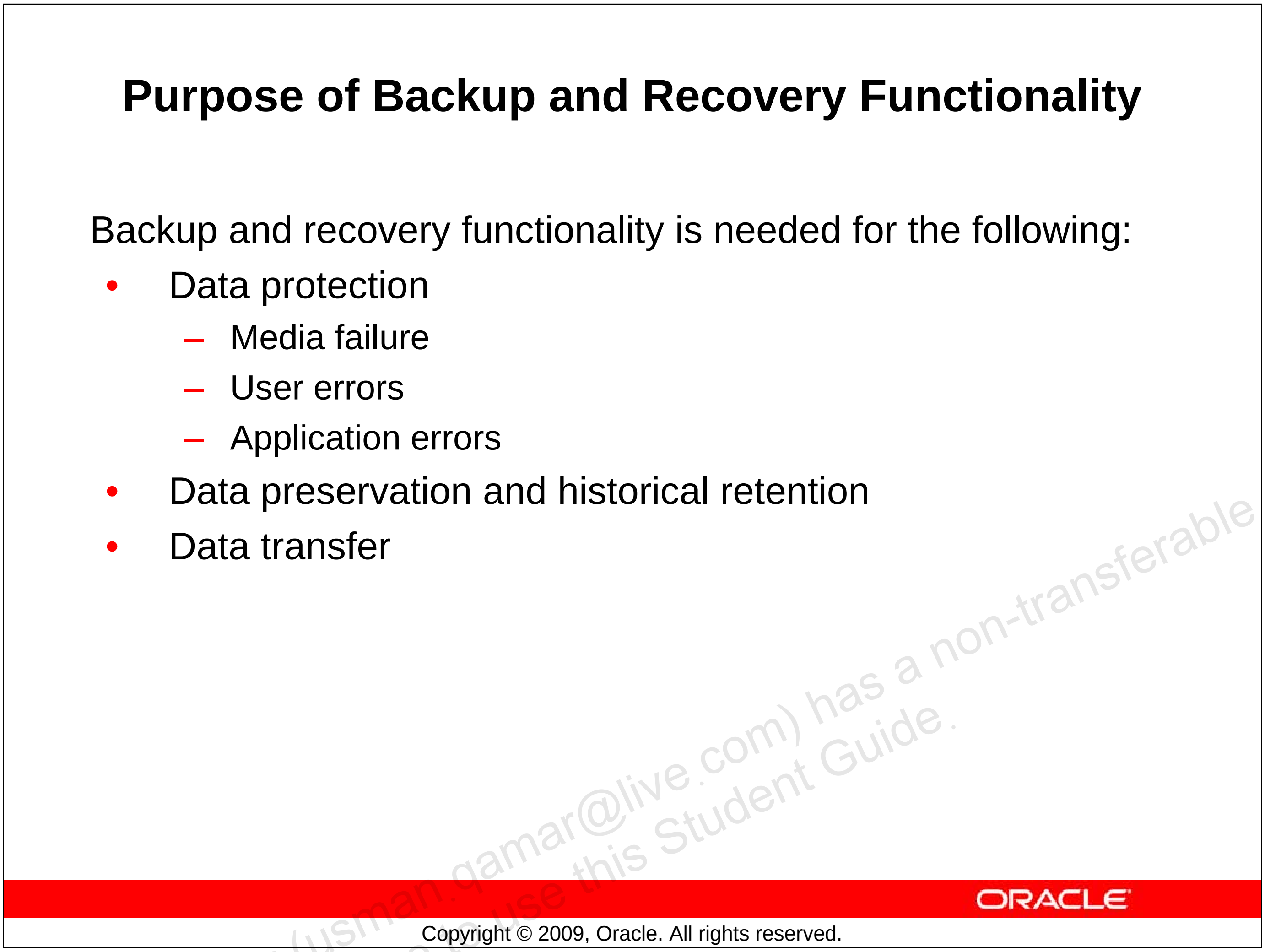
Objectives

After completing this lesson, you should be able to:

- Invoke and configure Recovery Manager (RMAN)
- Configure your database in `ARCHIVELOG` mode
- Configure multiple archive log file destinations to increase availability
- Configure the Fast Recovery Area (FRA)
- Specify a retention policy

Objectives

The Fast Recovery Area (FRA) was formerly known as Flash Recovery Area.



Purpose of Backup and Recovery Functionality

Backup and recovery functionality is needed for the following:

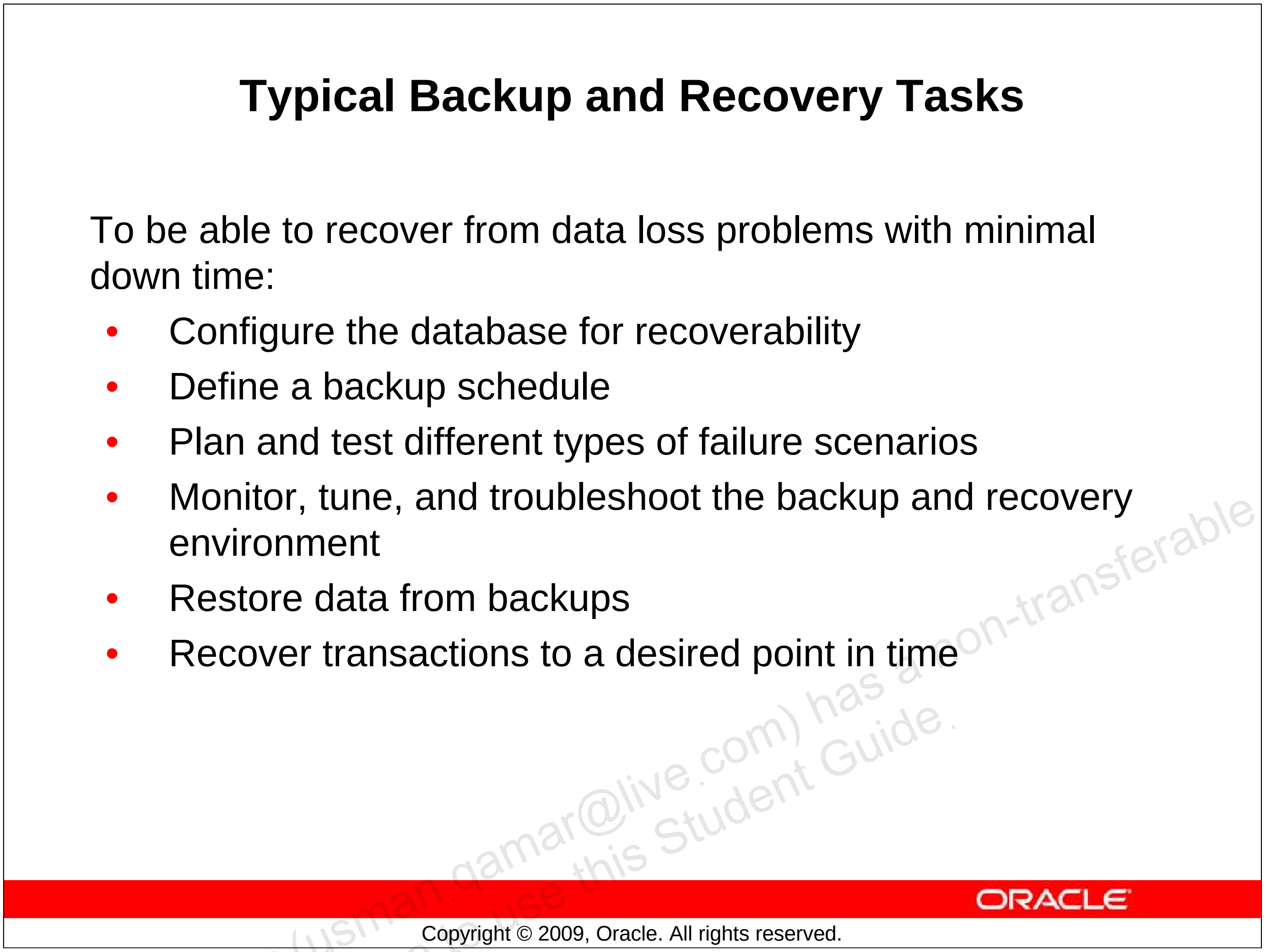
- Data protection
  - Media failure
  - User errors
  - Application errors
- Data preservation and historical retention
- Data transfer

Purpose of Backup and Recovery Functionality

The purpose of backup and recovery is to restore a failed database. Backups protect the database against problems such as hardware failure, media failure, user errors, and application errors. Media errors cause data problems by failing at the hardware level; a bad controller or disk drive can introduce either subtle or obvious errors. Users can also cause data errors, simply by issuing commands that should not be issued. Those same types of errors can be caused by an application with a bug.

Backups can also be used for data preservation and historical retention. Backups taken and preserved in `ARCHIVELOG` mode can be used to recover a database to a past point in time, which can be useful to meet compliance regulations.

You can also use backup and recovery tools to move data to other databases—even in other locations. A backup of a database is one possible way to duplicate it at another location.



Typical Backup and Recovery Tasks

To be able to recover from data loss problems with minimal down time:

- Configure the database for recoverability
- Define a backup schedule
- Plan and test different types of failure scenarios
- Monitor, tune, and troubleshoot the backup and recovery environment
- Restore data from backups
- Recover transactions to a desired point in time

Typical Backup and Recovery Tasks

A robust backup and recovery plan is important for a database that you cannot afford to lose. The plan includes the following tasks:

- **Configuration:** A backup and recovery strategy needs to be configured for your environment. This should include backup method, destination for backups, backup retention time and deletion of backups, and backup protection (encryption), if needed.
- **Scheduling:** Backups should be scheduled to run automatically during nonpeak hours.
- **Testing:** Periodically test backups and recovery practices.
- **Monitoring:** Monitor the effects of the backup operations to determine performance degradation on production databases and improve backup efficiency, when necessary.
- **Restoration:** A corrupted data file is overwritten from a backup of the data file. The data file is at a prior point of time than the current database.
- **Recovery:** Recovery applies the changes to the individual blocks, using archive and redo information, to move the database forward to the current point in time.

Oracle Backup and Recovery Solutions

For a recoverable system:

- RMAN
  - Block media recovery
  - Unused block compression
  - Binary compression
  - Backup encryption
- Solutions achieved via backup types:
  - All data blocks within your chosen files (full or incremental level 0)
  - Only information that has changed since a previous backup (incremental)
    - Cumulative (changes up to last level 0)
    - Differential (changes up to last incremental)

Oracle Backup and Recovery Solutions

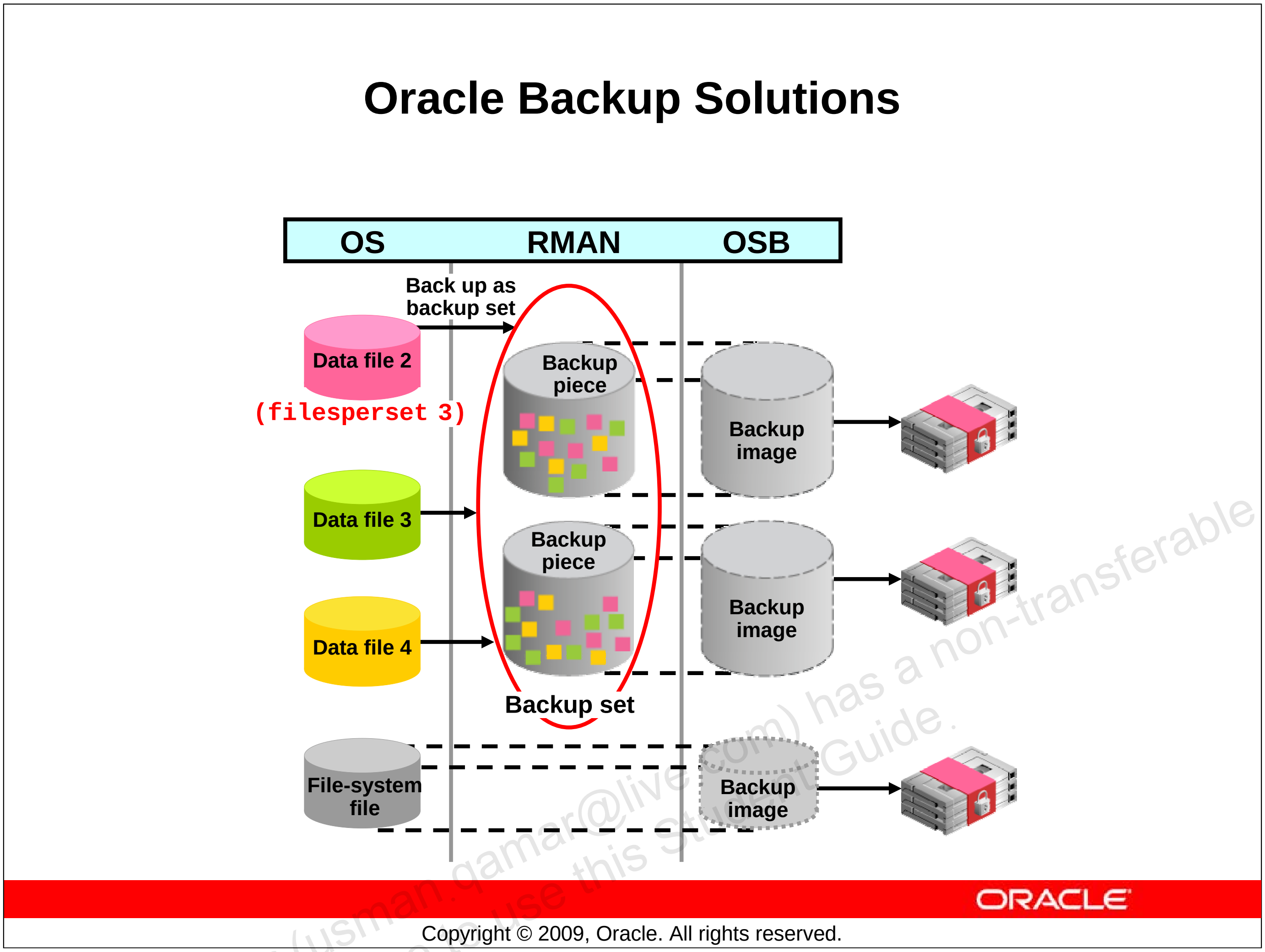
The following are major backup and recovery solutions:

Recovery Manager: A utility (with graphic and command-line interfaces) for performing backup and recovery. Some of the major features available when using RMAN are:

- **Block media recovery:** A method of recovering specific blocks of data, as opposed to entire tables (with Data Pump) or data files (with RMAN)
- **Unused block compression:** Unused block compression is a method by which blocks that are not currently used by the database are not read by the backup, and thus, not included in the backup.
- **Binary compression:** A space-saving feature in which backup files are compressed using well-known algorithms (comparable to utilities such as `zip` on Linux)
- **Backup encryption:** A security device for protecting backups you make

The solutions are achieved with these backup types:

- **Full backup:** Makes a copy of each data block that contains data and that is within the files being backed up
- **Incremental backup:** Makes a copy of all data blocks that have changed since a previous backup. The Oracle database supports two levels of incremental backup (0 and 1). A level 1 incremental backup can be one of two types: *cumulative* or *differential*. A cumulative backup backs up all changes since the last level 0 backup. A differential backup backs up all changes since the last incremental backup (which could be either a level 0 or level 1 backup).



Oracle Backup Solutions

OS RMAN OSB

Back up as backup set

Data file 2

(filesperset 3)

Data file 3

Data file 4

File-system file

Backup piece

Backup piece

Backup set

Backup image

Backup image

Backup image

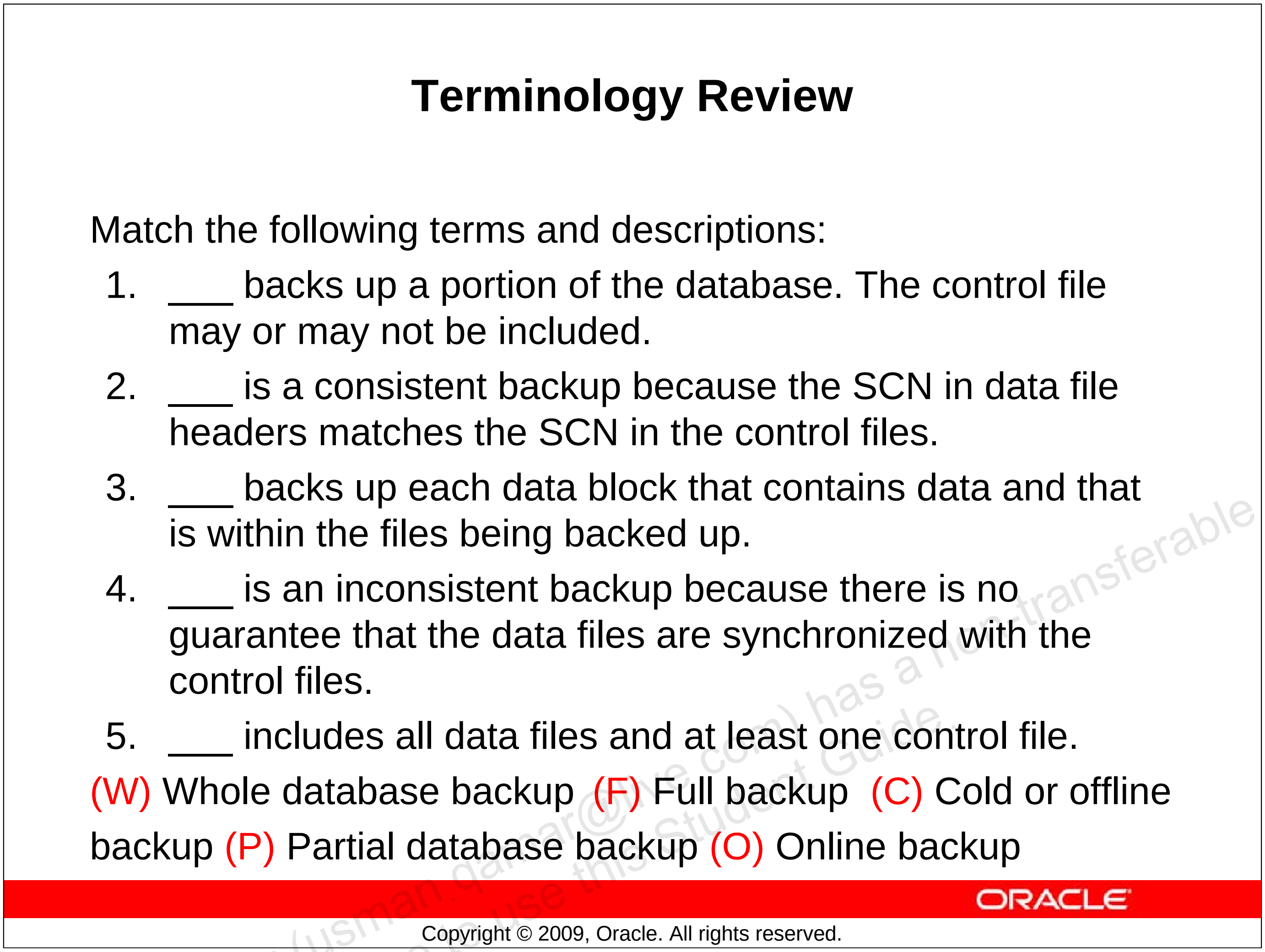
Oracle Backup Solutions

The slide shows backup sets. On the left are data files at the OS level. You see how they relate to the RMAN image copies and backup pieces (middle section) and how these relate to the Oracle Secure Backup (OSB) backup images.

At the bottom of the diagram are file-system files, which do not have an RMAN equivalent; they relate directly to OSB backup images.

- RMAN backs up data files, control files, redo log archives, and SPFILEs, whether they are the originals, backup sets, or image copies. RMAN performs backup and recovery operations to disk, and with the help of a media management layer (MML) such as Oracle Secure Backup, also to and from tape.
- Oracle Secure Backup (OSB) is a centralized tape management software for your entire Oracle environment, including file systems and the Oracle database. OSB can back up and restore data locally or over the local area network (LAN), wide area network (WAN), or SAN.

This diagram shows a subset of the Oracle Backup solution. Not included are image copies and proxy copies, which may be supported by some media managers.



Terminology Review

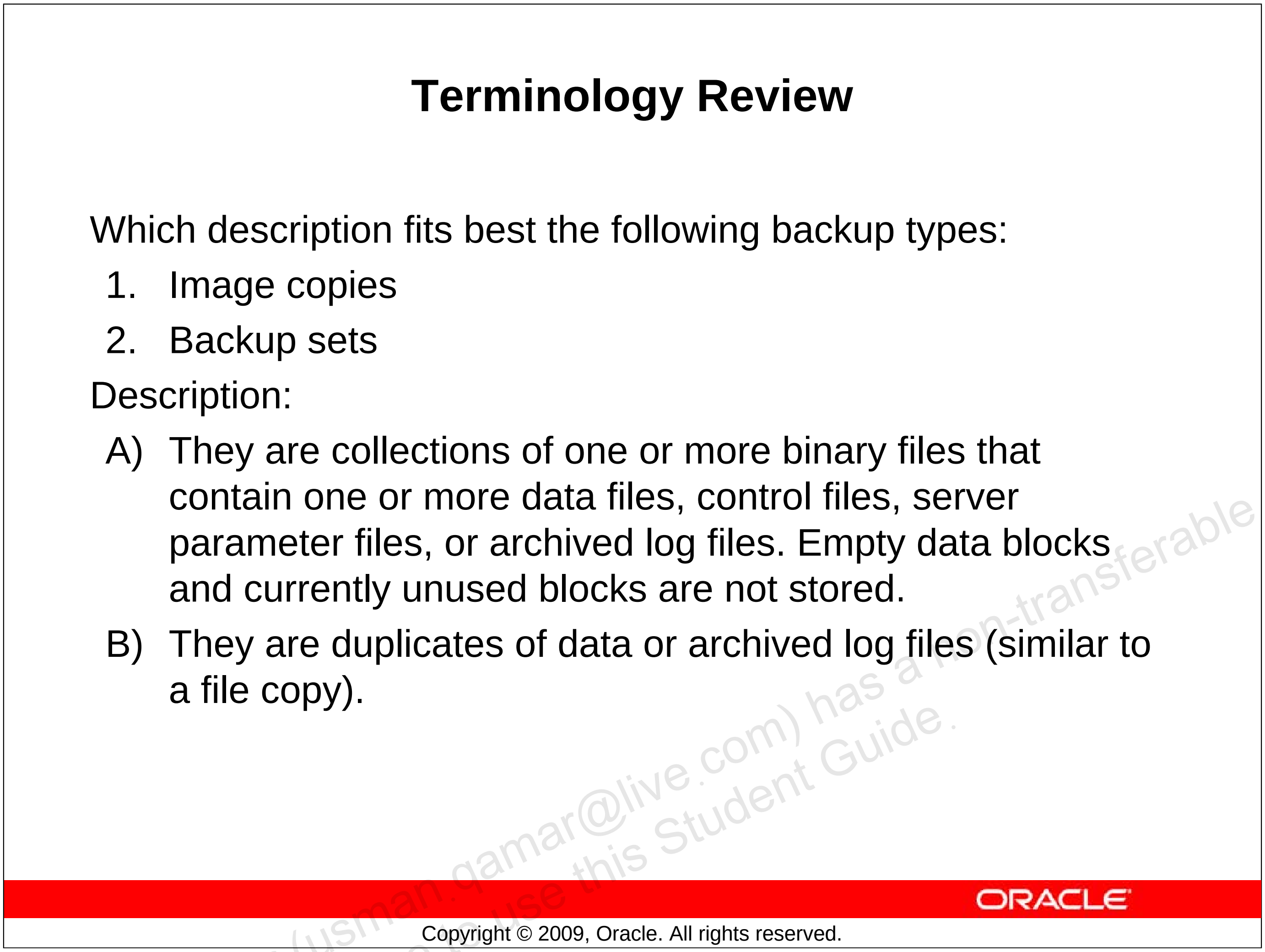
Match the following terms and descriptions:

1. ___ backs up a portion of the database. The control file may or may not be included.
2. ___ is a consistent backup because the SCN in data file headers matches the SCN in the control files.
3. ___ backs up each data block that contains data and that is within the files being backed up.
4. ___ is an inconsistent backup because there is no guarantee that the data files are synchronized with the control files.
5. ___ includes all data files and at least one control file.

(W) Whole database backup (F) Full backup (C) Cold or offline backup (P) Partial database backup (O) Online backup

Terminology Review

Correct Answers: 1P, 2C, 3F, 4O, 5W



Terminology Review

Which description fits best the following backup types:

1. Image copies
2. Backup sets

Description:

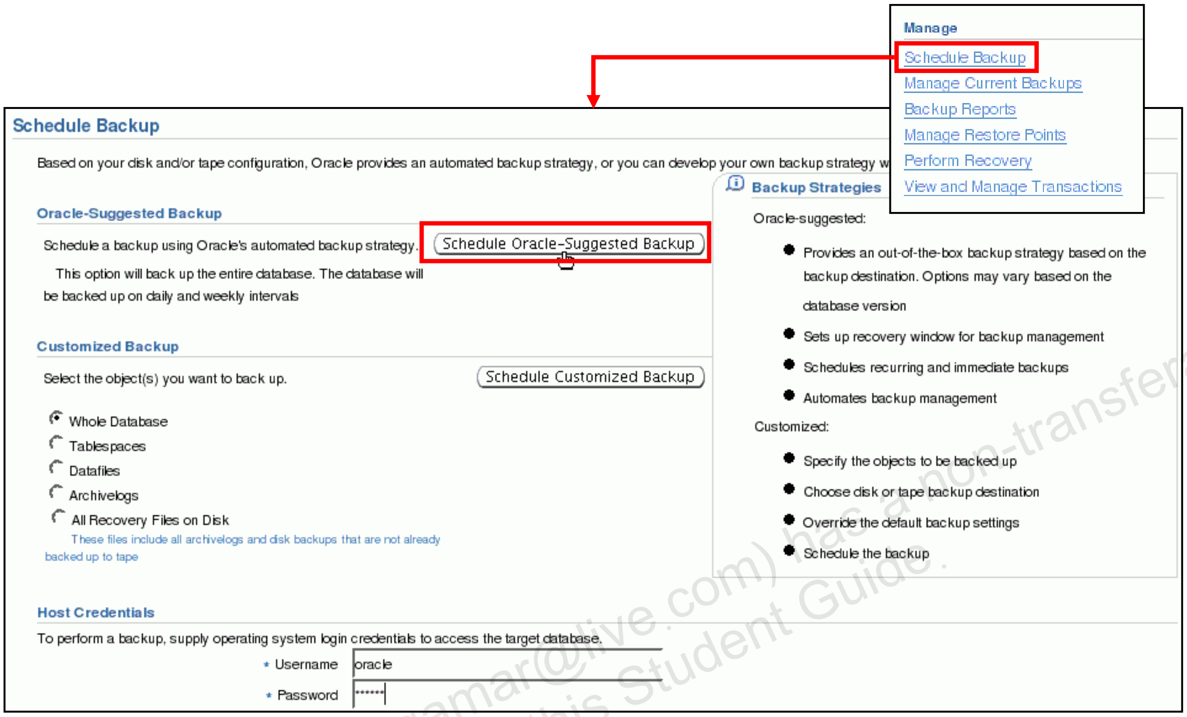
A) They are collections of one or more binary files that contain one or more data files, control files, server parameter files, or archived log files. Empty data blocks and currently unused blocks are not stored.

B) They are duplicates of data or archived log files (similar to a file copy).

Terminology Review (continued)

Correct Answers: 1B, 2A

What You Already Know: Oracle-Suggested Backup



Manage

- Schedule Backup
- Manage Current Backups
- Backup Reports
- Manage Restore Points
- Perform Recovery
- View and Manage Transactions

Schedule Backup

Based on your disk and/or tape configuration, Oracle provides an automated backup strategy, or you can develop your own backup strategy w

Oracle-Suggested Backup

Schedule a backup using Oracle's automated backup strategy. [Schedule Oracle-Suggested Backup]

This option will back up the entire database. The database will be backed up on daily and weekly intervals

Customized Backup

Select the object(s) you want to back up. [Schedule Customized Backup]

● Whole Database
○ Tablespaces
○ Datafiles
○ Archivelogs
○ All Recovery Files on Disk
These files include all archivelogs and disk backups that are not already backed up to tape

Backup Strategies

Oracle-suggested:

- Provides an out-of-the-box backup strategy based on the backup destination. Options may vary based on the database version
- Sets up recovery window for backup management
- Schedules recurring and immediate backups
- Automates backup management

Customized:

- Specify the objects to be backed up
- Choose disk or tape backup destination
- Override the default backup settings
- Schedule the backup

Host Credentials

To perform a backup, supply operating system login credentials to access the target database.

* Username: oracle
* Password: ••••••

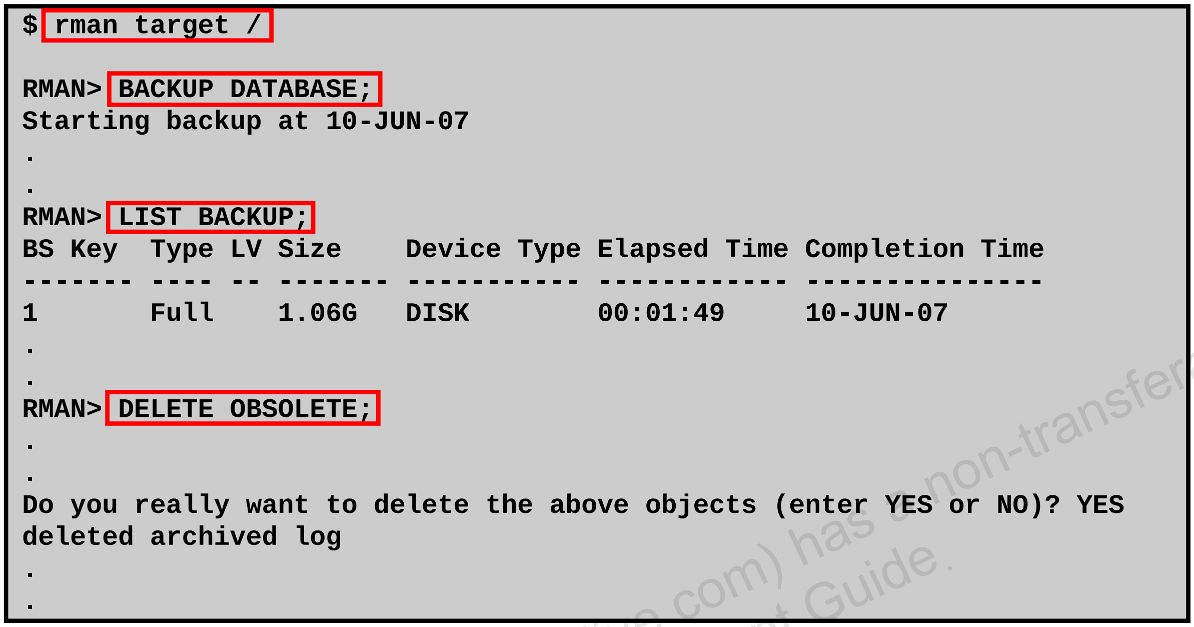
What You Already Know: Oracle-Suggested Backup

Enterprise Manager makes it easy for you to set up an Oracle-suggested backup strategy that protects your data and provides efficient recoverability to any point in the preceding 24 hours, and possibly as far back as 48 hours, depending on when the last backup was created. The Oracle-suggested strategy uses the incremental backup and incrementally updated backup features, providing faster recoverability than is possible when applying database changes from the archived log files.

To establish an Oracle-suggested strategy, navigate to the Maintenance page. In the Backup/Recovery region, select Schedule Backup. The Backup Strategies section enables you to select from the Oracle-suggested backup and Customized backup strategies. The Oracle-suggested strategy takes a full database copy as the first backup. Because it is a whole database backup, you might want to consider taking this at a period of least activity. After that, an incremental backup to disk is taken every day. Optionally, a weekly tape backup can be made, which backs up all recovery-related files.

Because these backups on disk are retained, you can always perform a full database recovery or a point-in-time recovery to any time within the past 24 hours, at the minimum. The recovery time could reach back as far as 48 hours. This is because just before a backup is taken on a given day, the backup from the *beginning* of day *n*–1 still exists.

Using Recovery Manager



```
$ rman target /

RMAN> BACKUP DATABASE;
Starting backup at 10-JUN-07
.
.
RMAN> LIST BACKUP;
BS Key  Type LV Size    Device Type Elapsed Time Completion Time
------- ---- -- ------- ----------- ------------ ---------------
1       Full    1.06G   DISK        00:01:49     10-JUN-07
.
.
RMAN> DELETE OBSOLETE;
.
.
Do you really want to delete the above objects (enter YES or NO)? YES
deleted archived log
.
.
```

Using Recovery Manager

Invoke RMAN at the operating system command line and specify the appropriate options. The following are the most commonly used options:

- **`target`:** The connect-string for the target database
- **`catalog`:** The connect-string for a recovery catalog
- **`nocatalog`:** Specifies there is no recovery catalog. This is the default.
- **`cmdfile`:** The name of an input command file
- **`log`:** The name of the output message log file

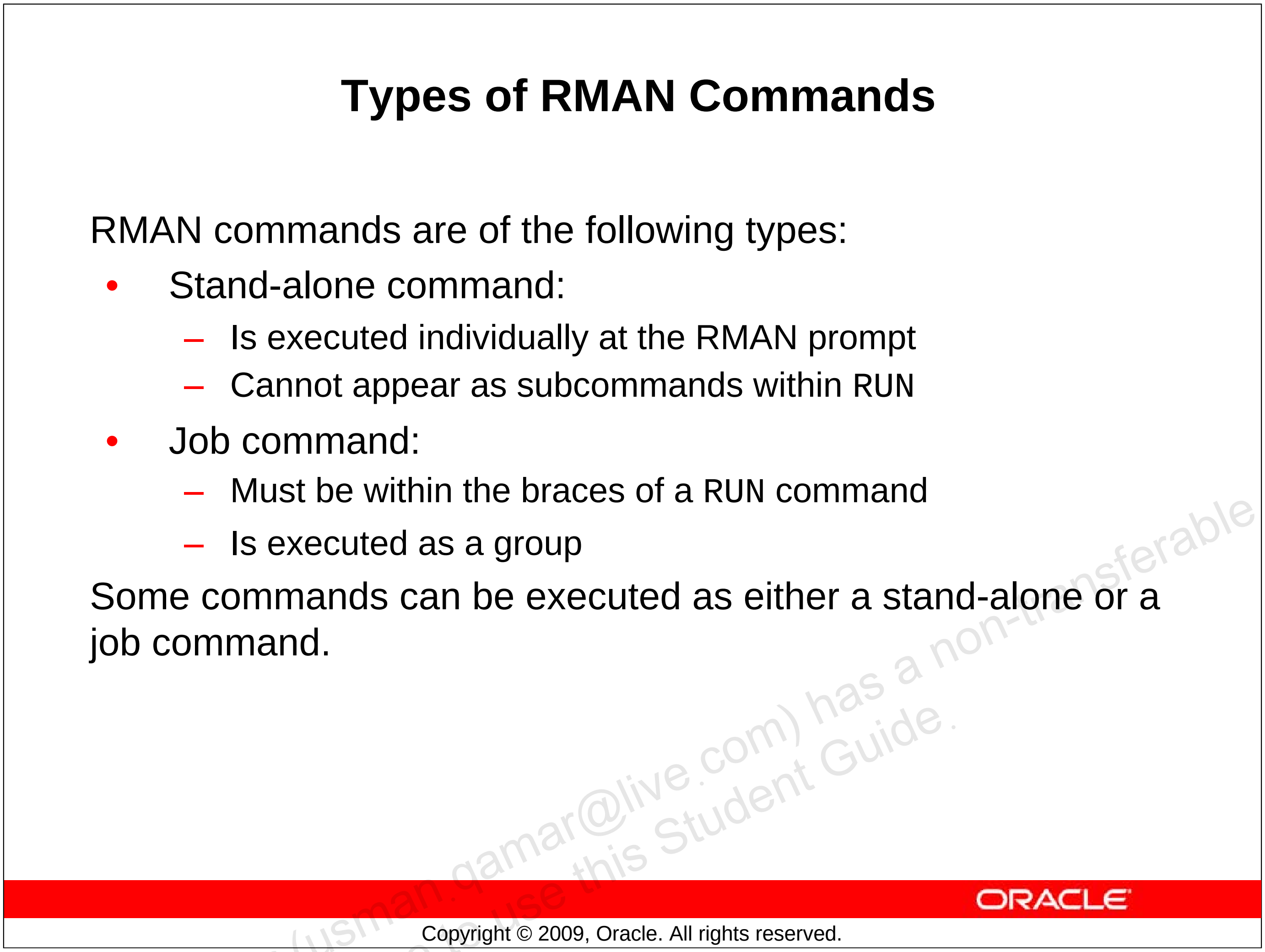
The RMAN invocation shown in the slide simply connects to the local database, as the target.

Here is an example of an RMAN invocation that connects to the local database using OS authentication, and specifies a command file to be run and a log file to receive a transcript of the RMAN commands that belong to the session:

```
        $ rman target / cmdfile=~/fullbu.rman log=~/fullbu.log
```

At the RMAN prompt, you can submit RMAN commands to manage your backup environment and create backups in many different ways, depending on your needs. Shown in the slide are commands to list the existing backups (`LIST BACKUP`) and delete any obsolete backups (`DELETE OBSOLETE`). The specifics of what these and other commands do are covered throughout this course.

Note: Refer to the *Oracle Database Backup and Recovery User's Guide* for more information about how to invoke RMAN. Refer to the *Oracle Database Backup and Recovery Reference* for the complete list of RMAN commands and their options.



Types of RMAN Commands

RMAN commands are of the following types:

- Stand-alone command:
  - Is executed individually at the RMAN prompt
  - Cannot appear as subcommands within RUN
- Job command:
  - Must be within the braces of a RUN command
  - Is executed as a group

Some commands can be executed as either a stand-alone or a job command.

Types of RMAN Commands

You can issue two basic types of RMAN commands: stand-alone and job commands.

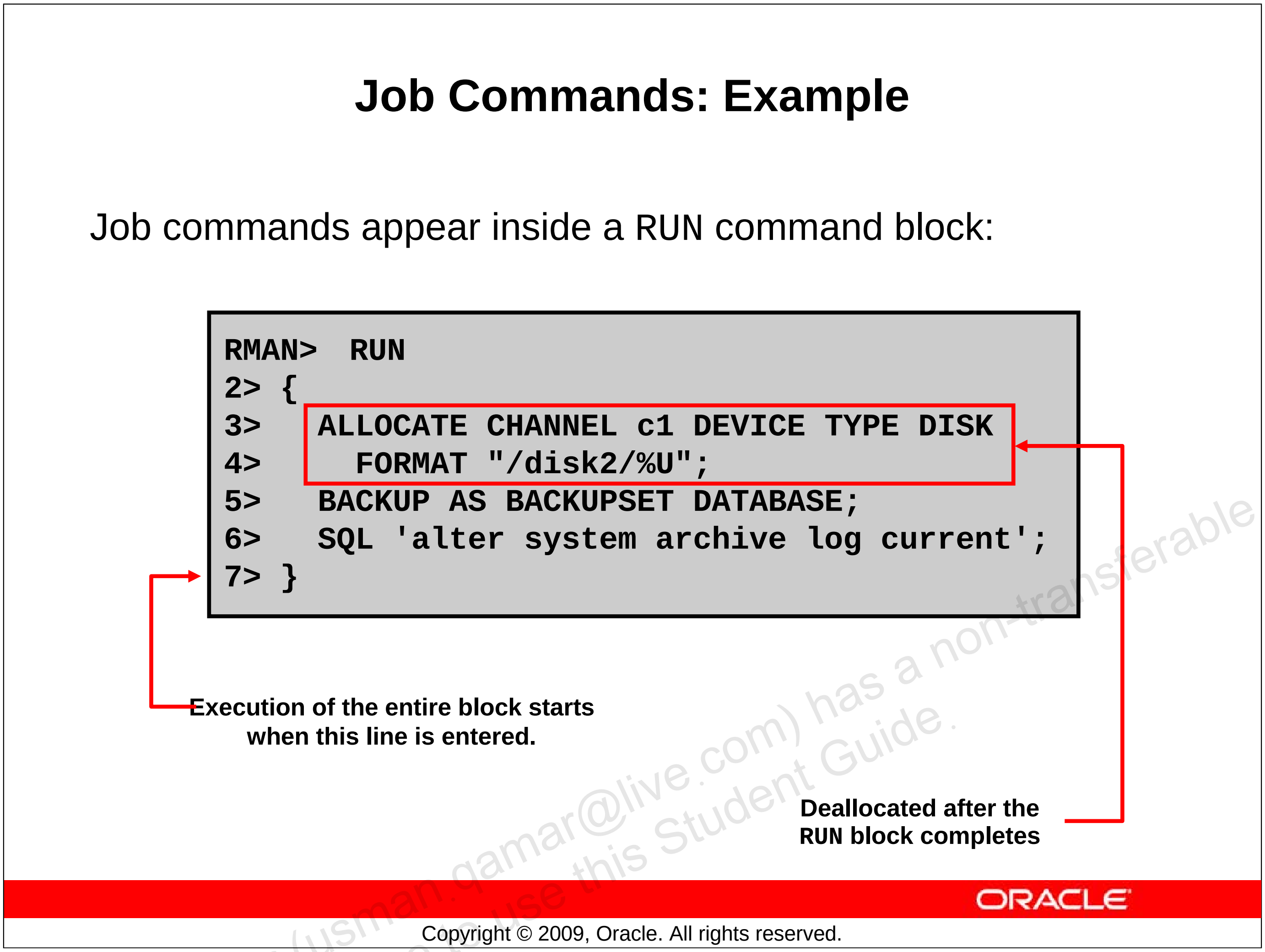
Stand-alone commands are executed at the RMAN prompt and are generally self-contained. Some of the stand-alone commands are:

- `CHANGE`
- `CONNECT`
- `CREATE CATALOG`, `RESYNC CATALOG`
- `CREATE SCRIPT`, `DELETE SCRIPT`, `REPLACE SCRIPT`

Job commands are usually grouped and executed sequentially inside a command block. If any command within the block fails, RMAN ceases processing; no further commands within the block are executed. The effects of any already executed commands still remain, though; they are not undone in any way.

An example of a command that can be run only as a job command is `ALLOCATE CHANNEL`. The channel is allocated only for the execution of the job, so it cannot be issued as a stand-alone command. There are some commands that can be issued either at the prompt or within a `RUN` command block, such as `BACKUP DATABASE`. If you issue stand-alone commands, RMAN allocates any needed channels by using the automatic channel allocation feature.

You can execute stand-alone and job commands in interactive mode or batch mode.



Job Commands: Example

Job commands appear inside a RUN command block:

```
RMAN>  RUN
2> {
3>   ALLOCATE CHANNEL c1 DEVICE TYPE DISK
4>     FORMAT "/disk2/%U";
5>   BACKUP AS BACKUPSET DATABASE;
6>   SQL 'alter system archive log current';
7> }
```

Execution of the entire block starts when this line is entered.

Deallocated after the RUN block completes

Job Commands: Example

Unlike stand-alone commands, job commands must appear within the braces of a RUN command. Commands placed inside a RUN block as shown in the slide are run as a single unit of commands. Any configurations made within the run block apply within the scope of the block and override any previously made settings. The following are examples of job commands, which must appear inside a RUN block:

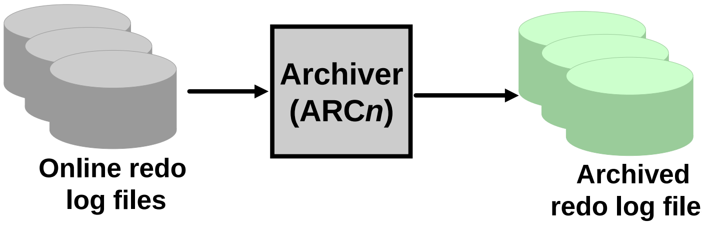
- ALLOCATE CHANNEL
- SWITCH

RMAN executes the job commands inside a RUN command block sequentially. If any command within the block fails, then RMAN ceases processing; no further commands within the block are executed. In effect, the RUN command defines a unit of command execution. When the last command within a RUN block completes, the Oracle database releases any server-side resources such as input/output (I/O) buffers or I/O slave processes allocated within the block.

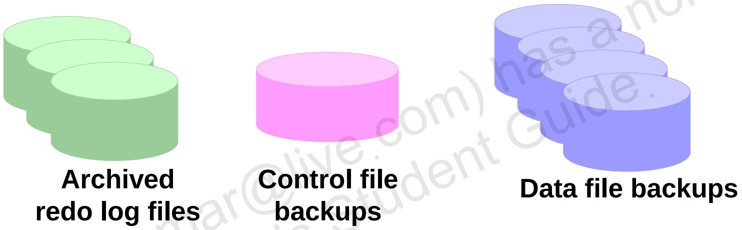
Note: The SQL command on line 6 is just an example. It is NOT a required command for the backup operation.

Configuring Your Database for Backup and Recovery Operations

- Operate the database in `ARCHIVELOG` mode.



- Configure the FRA.



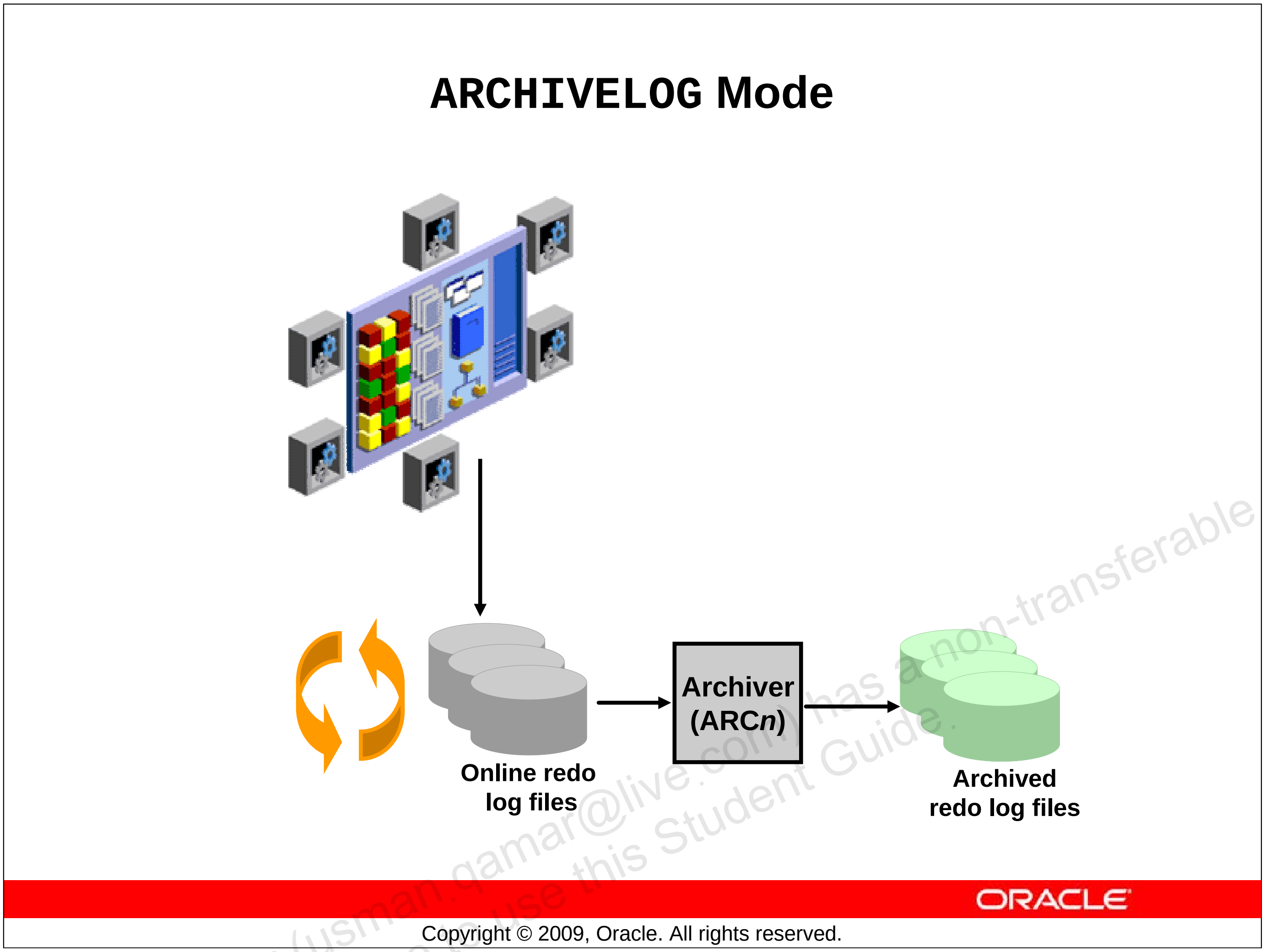
Configuring Your Database for Backup and Recovery Operations

When you operate your database in `ARCHIVELOG` mode, you have more recovery options after a data loss, including point-in-time recovery of the database or some tablespaces.

It is recommended that you take advantage of the Fast Recovery Area to store as many backup and recovery–related files as possible, including disk backups and archived redo logs.

Some features of Oracle Database backup and recovery, such as Oracle Flashback Database and guaranteed restore points, require the use of a Fast Recovery Area.

Both of the features are covered in detail later in this lesson.



ARCHIVELOG Mode

Online redo log files

Archiver (ARC*n*)

Archived redo log files

ORACLE

ARCHIVELOG Mode

As modifications to data in the database are made, the redo data is written out to the online redo log file. A given file is specified as being written to at a given time. When it is full, the Archiver process (ARC*n*) copies the online log file to another location that serves as an archive of that file, which can be preserved for as long as you need it. This provides more opportunities for recovery, because you can save, back up, and restore all of the archive redo logs ever generated.

Because the online redo log files are reused in a circular fashion, there is a protocol for controlling when one is allowed to be reused. In `ARCHIVELOG` mode, the database only begins writing to an online redo log file if it has been archived. This ensures that every redo log file has a chance to be archived.

Configuring ARCHIVELOG Mode

To place the database in ARCHIVELOG mode, perform the following steps:

- Using Enterprise Manager
  - Select the “ARCHIVELOG Mode” check box.
  - Click Apply. The database can be set to ARCHIVELOG mode only from the MOUNT state.
  - Click Yes when asked whether you want to restart the database.
- Using SQL commands
  - Mount the database.
  - Issue the ALTER DATABASE ARCHIVELOG command.
  - Open the database.

Configuring ARCHIVELOG Mode

Placing the database in ARCHIVELOG mode prevents redo logs from being overwritten until they have been archived.

In Enterprise Manager, do this by navigating to Availability > Recovery Settings and selecting the ARCHIVELOG Mode check box. The database must be restarted after making this change.

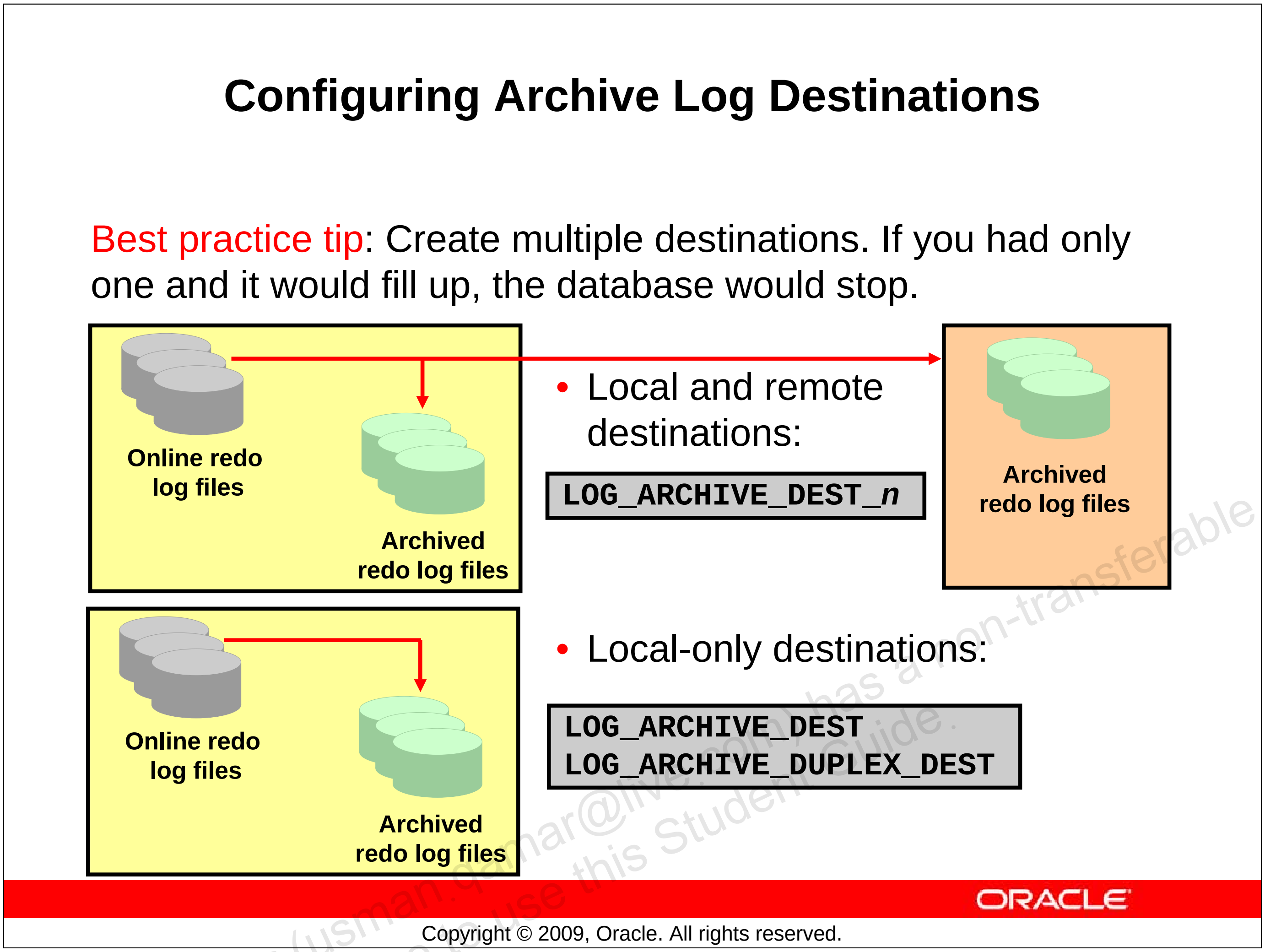
To issue the SQL command to put the database in ARCHIVELOG mode, the database must be in MOUNT mode. If the database is currently open, you must shut it down cleanly (not abort), and then mount it. The following shows the commands to shut down an open database, put it in ARCHIVELOG mode, and then open it:

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```

With the database in NOARCHIVELOG mode (the default), recovery is possible only until the time of the last backup. All transactions made after that backup are lost.

In ARCHIVELOG mode, recovery is possible until the time of the last commit. Most production databases are operated in ARCHIVELOG mode.

Note: Back up your database after switching to ARCHIVELOG mode because your database is recoverable only from the first backup taken in that mode.



Configuring Archive Log Destinations

Best practice tip: Create multiple destinations. If you had only one and it would fill up, the database would stop.

Online redo log files

Archived redo log files

- Local and remote destinations:

`LOG_ARCHIVE_DEST_n`

Archived redo log files

Online redo log files

Archived redo log files

- Local-only destinations:

```
LOG_ARCHIVE_DEST
LOG_ARCHIVE_DUPLEX_DEST
```

Configuring Archive Log Destinations

Best practice tip: You should create multiple archive log destinations, because if you had only one destination and it would fill up, then your database would stop.

Local and remote destinations: Specify local and remote destinations by setting the set of `LOG_ARCHIVE_DEST_n` initialization parameters. There are ten of these, so *n* can be 1 through 10.

- In order to specify a local storage location, supply a local directory name for the value of one of these variables, supplying the "`LOCATION=`" string. For example, to specify the `/disk3/arch` directory, set one of these variables as follows:

  ```
  LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk3/arch‘
  ```

- If you want to specify a remote location for a standby database, use the `SERVICE` keyword in the value, as in the following example, where `standyby1` is the network service name for the standby database instance:

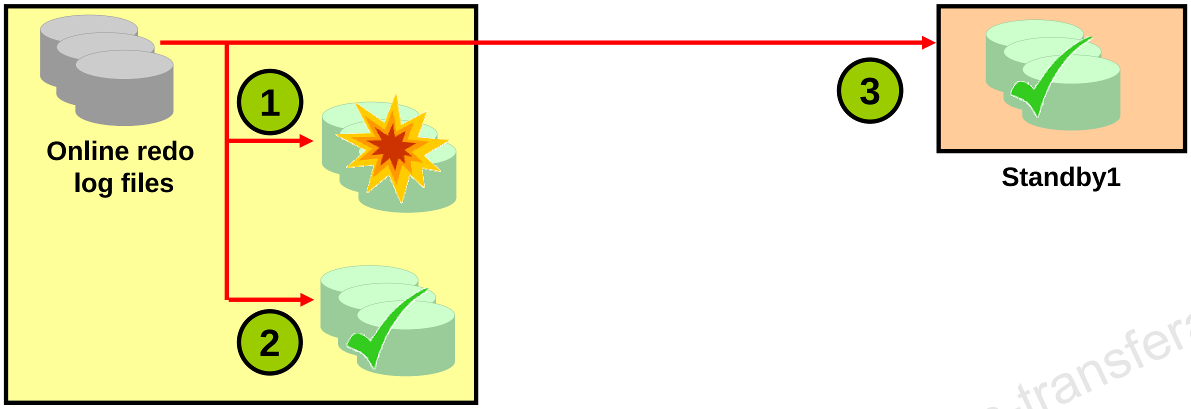
  ```
  LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1'.
  ```

Local-only destinations: With the Standard edition of the Oracle database, set the `LOG_ARCHIVE_DEST` and the `LOG_ARCHIVE_DUPLEX_DEST` parameters to local disk directories. Thus, you can have up to two archive log file locations. For example:

```
LOG_ARCHIVE_DEST = '/disk1/arch'
LOG_ARCHIVE_DUPLEX_DEST = '/disk2/arch‘
```

Where available, Oracle recommends that you use the `LOG_ARCHIVE_DEST_n` method, because this allows the most flexibility in the type and the number of destinations.

Guaranteeing Archive Log Success



```
LOG_ARCHIVE_MIN_SUCCEED_DEST = 2
```

Guaranteeing Archive Log Success

If you have specified more than one destination for archive log files, you should specify a minimum number of them that are required to succeed in order for the archive to be considered successful. Do this using the `LOG_ARCHIVE_MIN_SUCCEED_DEST` initialization parameter. Set it to the number of destinations that must succeed in receiving the archived log file. The online log file is not reused until this number is met.

In the example in the slide, there are three destinations specified: two are local, and one is remote. `LOG_ARCHIVE_MIN_SUCCEED_DEST` is set to 2, which means as long as at least two of the destinations succeed, the online redo log file can be overwritten. The example shows that destination 1 has failed. That does not stop the database, because two of them succeeded.

You can use this parameter with either of the models described in the previous slide. If you use it with the `LOG_ARCHIVE_DEST_`*n* model, then this parameter can have values ranging from 1 through 10. If you use it with the `LOG_ARCHIVE_DEST` model, then the values can be 1 or 2, because you can specify only two destinations in that case.

Guaranteeing Archive Log Success (continued)

Specifying MANDATORY and OPTIONAL

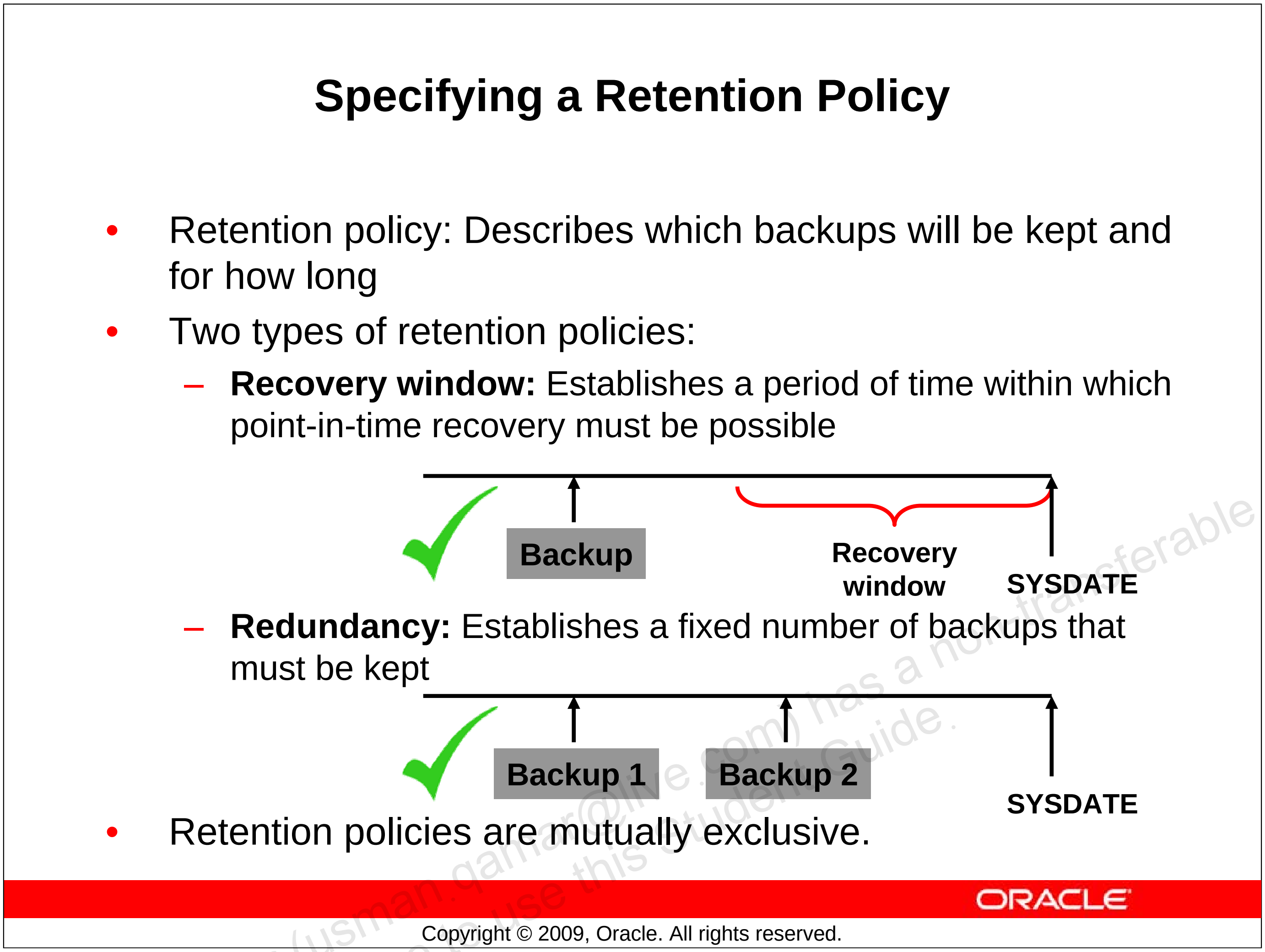
When you define a destination, you can specify that it is a mandatory one. Do this by specifying the MANDATORY or OPTIONAL keyword after the location specification. Here is an example:

```
        LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk3/arch MANDATORY'
```

The default is OPTIONAL.

A mandatory destination is given special consideration. If any mandatory destination fails, then Oracle Database considers the archiving of the log to have not succeeded, and the online redo log file is not allowed to be overwritten. In this case, it ignores the LOG_ARCHIVE_MIN_SUCCEED_DEST parameter.

Any destination specified by LOG_ARCHIVE_DEST is mandatory. Any destination declared by LOG_ARCHIVE_DUPLEX_DEST is optional if LOG_ARCHIVE_MIN_SUCCEED_DEST = 1 and mandatory if LOG_ARCHIVE_MIN_SUCCEED_DEST = 2.



Specifying a Retention Policy

- Retention policy: Describes which backups will be kept and for how long
- Two types of retention policies:
  - **Recovery window:** Establishes a period of time within which point-in-time recovery must be possible
  - **Redundancy:** Establishes a fixed number of backups that must be kept
- Retention policies are mutually exclusive.

Specifying a Retention Policy

A *retention policy* describes which backups will be kept and for how long. You can set the value of the retention policy by using the RMAN `CONFIGURE` command or Enterprise Manager.

Recovery Window Retention Policy

The best practice is to establish a period of time during which it will be possible to discover logical errors and fix the affected objects by doing a point-in-time recovery to just before the error occurred. This period of time is called the *recovery window*. This policy is specified in number of days. For each data file, there must always exist at least one backup that satisfies the following condition:

```
        SYSDATE – backup_checkpoint_time >= recovery_window
```

You can use the following command syntax to configure a recovery window retention policy:

```
        RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <days>
        DAYS;
```

where <*days*> is the size of the recovery window.

If you are not using a recovery catalog, then you should keep the recovery window time period less than or equal to the value of the control file parameter `CONTROL_FILE_RECORD_KEEP_TIME` to prevent the record of older backups from being overwritten in the control file. If you are using a recovery catalog, then make sure `CONTROL_FILE_RECORD_KEEP_TIME` is greater than the time period between catalog resynchronizations. Resynchronizations happen when you:

- Create a backup. In this case the synchronization is done implicitly.
- Execute the `RESYNC CATALOG` command

Specifying a Retention Policy (continued)

Recovery catalogs are covered in more detail in the lesson titled “Using the RMAN Recovery Catalog.”

Redundancy Retention Policy

If you require a certain number of backups to be retained, then you can set the retention policy on the basis of the redundancy option. This option requires that a specified number of backups be cataloged before any backup is identified as obsolete. The default retention policy has a redundancy of 1, which means that only one backup of a file must exist at any given time. A backup is deemed obsolete when a more recent version of the same file has been backed up.

You can use the following command to reconfigure a redundancy retention policy:

```
        RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY <copies>;
```

where <*copies*> is the number of copies that are required for policy satisfaction.

Disabling the Retention Policy

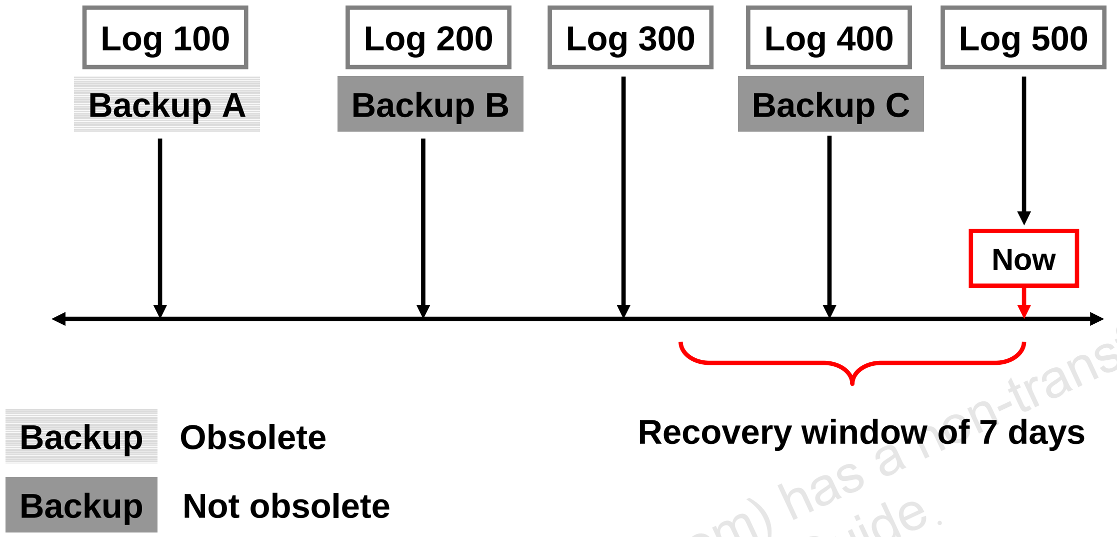
You may want to disable the retention policy totally. If you have a separate system, outside of RMAN, that backs up your disk backups to tape, you may want to do this. If you disable the retention policy, then RMAN never considers a backup obsolete. Because RMAN does not have to decide when to remove a backup from disk (because another utility is managing that), RMAN does not need to be configured for making that decision. In this case, records of each backup are maintained for as long as is specified by the `CONTROL_FILE_RECORD_KEEP_TIME` initialization parameter.

Disable the retention policy by using this command:

```
        RMAN> CONFIGURE RETENTION POLICY TO NONE;
```

Note: You can specify that a backup is an exception to the retention policy that you have defined. This is called an archival backup, which is covered in the lesson titled “Using RMAN to Create Backups.”

A Recovery Window Retention Policy: Example



Log 100 | Log 200 | Log 300 | Log 400 | Log 500

Backup A | Backup B | Backup C

Now

Recovery window of 7 days

Backup	Obsolete
Backup	Not obsolete

Backup B and archive logs 201 through 500 are required to satisfy this retention policy.

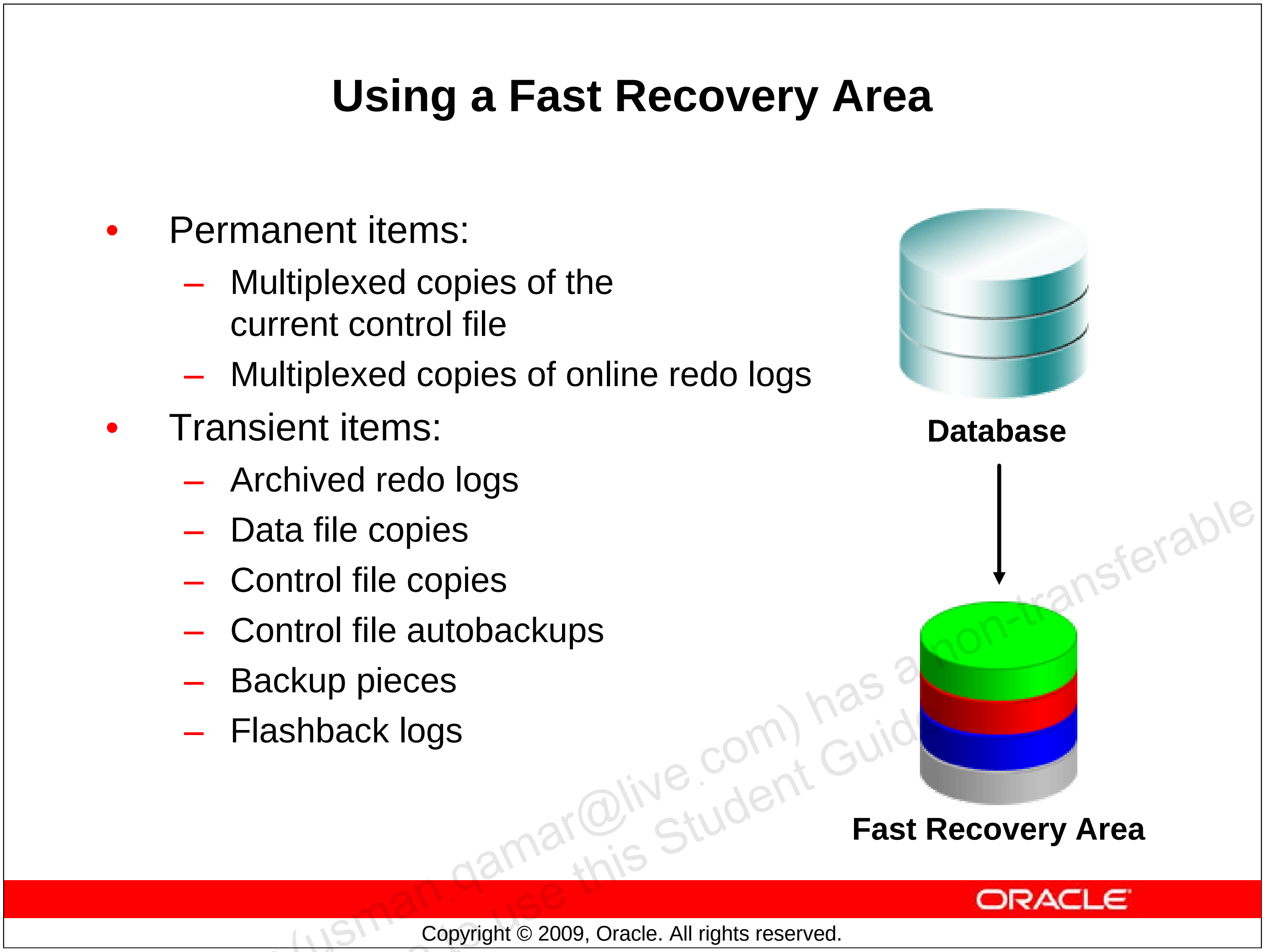
ORACLE

Specifying a Recovery Window Retention Policy: Example

The retention policy in the slide shows that it requires the ability to recover to any time within the last seven days. Some of the backups and logs are obsolete, because they are not needed to recover to a time within the seven-day window. This retention policy is configured thus:

```
        RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
```

Given the backups and archived log files available, the only data needed to recover to a point inside the recovery window is Backup B and logs 201 through 500. Note that Backup A is not needed because there is a later backup (B) that is still before the recovery window. Also, Backup C is not sufficient as the only backup to retain because it would not satisfy a need to recover to points in time at the beginning of the recovery window. The last backup that was taken before the beginning of the recovery window, including all logs since that backup, is what is necessary.



Using a Fast Recovery Area

- Permanent items:
  - Multiplexed copies of the current control file
  - Multiplexed copies of online redo logs
- Transient items:
  - Archived redo logs
  - Data file copies
  - Control file copies
  - Control file autobackups
  - Backup pieces
  - Flashback logs

Using a Fast Recovery Area

The Fast Recovery Area is a unified storage location for all recovery-related files and activities in an Oracle database. All files that are needed to completely recover a database from a media failure are part of the Fast Recovery Area. The recovery-related files are of two types: permanent and transient. Permanent files are actively being used by the instance. Transient files are needed only in the event of some type of recovery operation.

Permanent Items

- **Control file:** Depending on the setting of several initialization parameters, a copy of the control file is created in the Fast Recovery Area location when you create a new database or control file. For details see the “Semantics” section of the `CREATE CONTROLFILE` command in the *Oracle Database SQL Language Reference*.
- **Multiplexed copies of online redo log files:** A mirrored copy from each redo log group can be here. When you create a database, you can specify the location of the online redo log files using the `LOGFILE` clause. If you do not include that clause, then the locations are set according to the values of the following initialization parameters:
  - **`DB_CREATE_ONLINE_LOG_DEST_`*`n`*`:`** If one or more of these variables is set, then these are the only locations used.
  - **`DB_CREATE_FILE_DEST:`** If this is set, then this is the primary file location.
  - **`DB_RECOVERY_FILE_DEST:`** If this is set, in addition to `DB_CREATE_FILE_DEST`, then this location is used as the mirror.

Using a Fast Recovery Area (continued)

For more details on how these variables affect the location of the online redo logs, see the `LOGFILE` clause of the `CREATE DATABASE` statement in the *Oracle Database SQL Language Reference*.

Transient Items

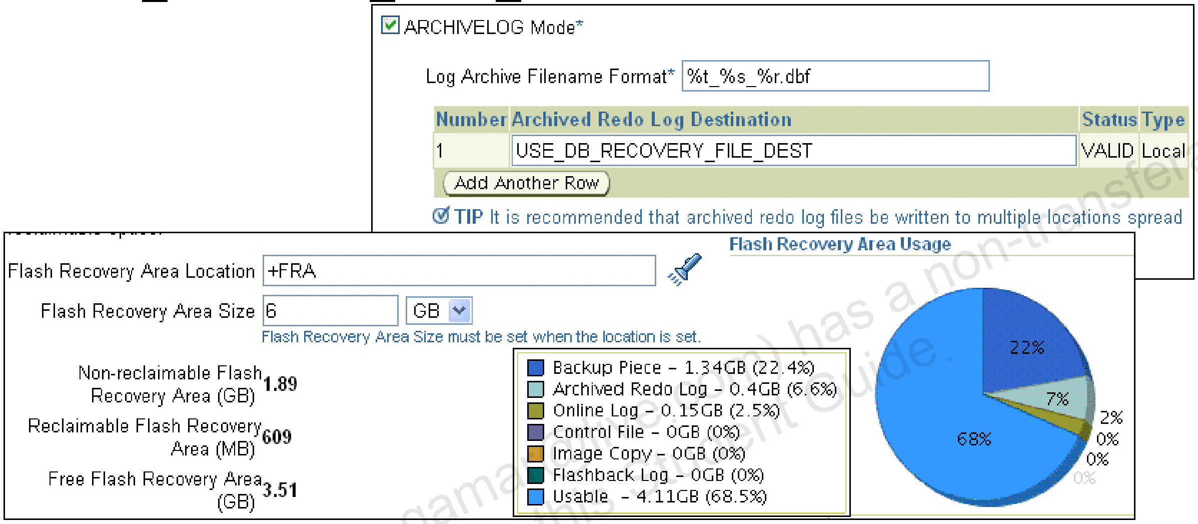
- **Archived redo log files:** When the Fast Recovery Area is configured, `LOG_ARCHIVE_DEST_1` is automatically set to the Fast Recovery Area location. The Archiver background process creates archived redo log files in the Fast Recovery Area and in other configured `LOG_ARCHIVE_DEST_`*n* locations. If no `LOG_ARCHIVE_DEST_`*n* locations are defined, the default location for archived redo log files is in the Fast Recovery Area.
- **Flashback logs:** Flashback logs are generated when Flashback Database is enabled.
- **Control file autobackups:** The default location for control file autobackups created by RMAN and autobackups generated by the Oracle database server is the Fast Recovery Area.
- **Data file copies:** The `BACKUP AS COPY` command creates image data file copies in the Fast Recovery Area.
- **RMAN files:** The Fast Recovery Area is the default location that is used by RMAN for backups and restoration of the archive log content from tape for a recovery operation.

Note: If you need a good performance for your FRA, consider creating it on its own physical disks and controllers.

Defining a Fast Recovery Area

The FRA is defined by setting two, dynamic initialization parameters:

- `DB_RECOVERY_FILE_DEST_SIZE`: Sets the disk limit
- `DB_RECOVERY_FILE_DEST`: Sets the location for the FRA



☑ ARCHIVELOG Mode*

Log Archive Filename Format* %t_%s_%r.dbf

Number	Archived Redo Log Destination	Status	Type
1	USE_DB_RECOVERY_FILE_DEST	VALID	Local

TIP It is recommended that archived redo log files be written to multiple locations spread

Flash Recovery Area Location +FRA

Flash Recovery Area Size 6 GB

Flash Recovery Area Size must be set when the location is set.

Non-reclaimable Flash Recovery Area (GB) 1.89

Reclaimable Flash Recovery Area (MB) 609

Free Flash Recovery Area (GB) 3.51

Flash Recovery Area Usage

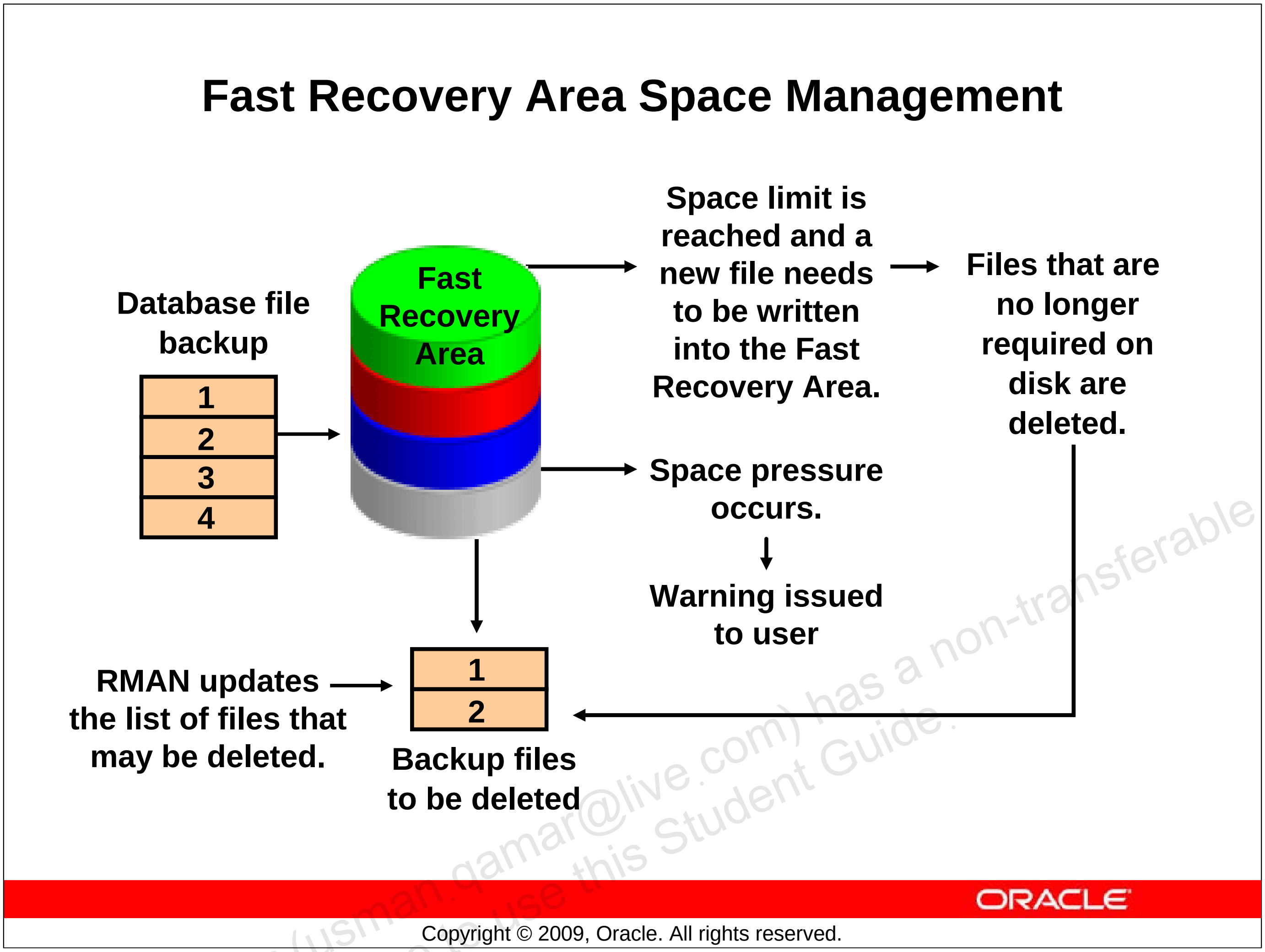
- Backup Piece - 1.34GB (22.4%)
- Archived Redo Log - 0.4GB (6.6%)
- Online Log - 0.15GB (2.5%)
- Control File - 0GB (0%)
- Image Copy - 0GB (0%)
- Flashback Log - 0GB (0%)
- Usable - 4.11GB (68.5%)

Defining a Fast Recovery Area

Use the following mandatory parameters to define the FRA:

- **`DB_RECOVERY_FILE_DEST_SIZE`:** You must define a disk limit, which is the amount of space that the FRA is permitted to use. Setting a limit allows the remaining disk space not dedicated to the FRA to be used for other purposes.
  - A basic recommendation for the size of the disk limit is the sum of the database size, the size of incremental backups, and the size of all archive log files that have not been copied to tape. The reason for this recommendation is that the Oracle-suggested backup strategy takes one image copy of the database (without temp files) and then the incremental backups.
  - The minimum size of the FRA should be at least large enough to contain archived redo log files that have not been copied to tape.
  - The sizing of the FRA is dependent on the backup strategy and other options that are implemented. Guaranteed Restore Points also have an effect on the size of the FRA.
- **`DB_RECOVERY_FILE_DEST`:** An FRA specification contains a location, which is a valid destination to create files.

You can use Enterprise Manager Grid Control and Database Control to easily define the FRA. Navigate to Availability > Recovery Settings. You can define the FRA location and its size on the Recovery Settings page. You must set the size of the FRA when specifying its location.



Fast Recovery Area Space Management

Database file backup

1
2
3
4

Fast Recovery Area

Space limit is reached and a new file needs to be written into the Fast Recovery Area.

Files that are no longer required on disk are deleted.

Space pressure occurs.

Warning issued to user

RMAN updates the list of files that may be deleted.

1
2

Backup files to be deleted

ORACLE

Fast Recovery Area Space Management

Each time RMAN creates a file in the Fast Recovery Area, the list of files that are no longer required on disk is updated. Based on the value of `DB_REOVERY_FILE_DEST_SIZE`, when the Fast Recovery Area experiences space pressure or is low on free space because there are no files that can be deleted from the Fast Recovery Area, you are warned of the danger of running out of space. The Oracle database server and RMAN continue to create files in the Fast Recovery Area until 100% of the disk limit is reached. When setting `DB_RECOVERY_FILE_DEST_SIZE`, you must allocate enough space to hold the recovery files, including backups that are waiting to be backed up to tape. Files that are obsolete or have been backed up to tape are likely candidates for deletion to provide free space.

When a file is written into the Fast Recovery Area and space is needed for that file, the Oracle database server deletes a file that is on the obsolete files list. When a file is written and deleted from the Fast Recovery Area, notification is written into the alert log.

Note: When the Fast Recovery Area’s used space is at 85%, a warning alert is issued, and when used space is at 97%, a critical alert is issued. These are internal settings and cannot be changed.

A sample alert log output follows:

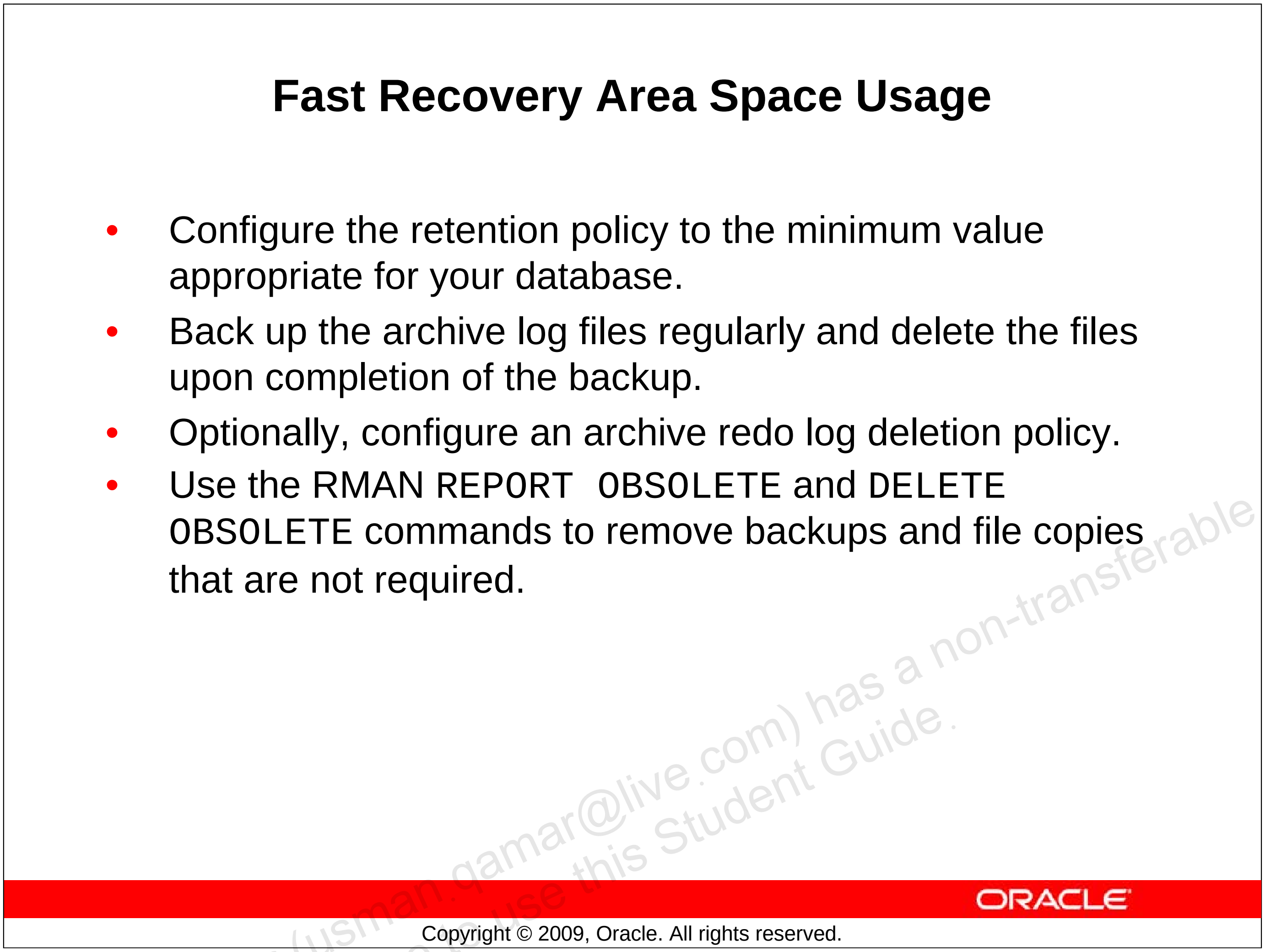
```
WARNING: db_recovery_file_dest_size of 52428800 bytes is 100.00%
used, and has 0 remaining bytes available.
```

Fast Recovery Area Space Management (continued)

You can issue the following query to determine the action to take:

```
SQL> SELECT object_type, message_type, message_level,
   2 reason, suggested_action
   3 FROM dba_outstanding_alerts;
```

Your choice is to add additional disk space, back up files to a tertiary device, delete files from the Fast Recovery Area using RMAN, or consider changing the RMAN retention policy.



Fast Recovery Area Space Usage

- Configure the retention policy to the minimum value appropriate for your database.
- Back up the archive log files regularly and delete the files upon completion of the backup.
- Optionally, configure an archive redo log deletion policy.
- Use the RMAN `REPORT OBSOLETE` and `DELETE OBSOLETE` commands to remove backups and file copies that are not required.

Fast Recovery Area Space Usage

To avoid running out of space in the Fast Recovery Area, perform the following steps as needed or appropriate:

- Use RMAN to delete unnecessary files from the Fast Recovery Area.
- Use RMAN to take frequent backups of the Fast Recovery Area.
- Change the RMAN retention policy to retain backups for a smaller period of time.
- Change the RMAN archived log deletion policy.
- Add disk space and increase the value of the `DB_RECOVERY_FILE_DEST_SIZE` database initialization parameter if you frequently run out of space.

Enterprise Manager does not report the amount of space used by the Fast Recovery Area on the disk or the amount of space used in the Fast Recovery Area directory tree, but it does report the sizes of the files that RMAN believes are in the directory. So do not put any files in this area that are not managed by RMAN.

If you remove any file from this area with a tool other than RMAN, use RMAN to remove the file entries from the catalog. For example, to back up the archived log files in the Fast Recovery Area and then delete the files after they have been successfully backed up, you would use the RMAN command as follows:

```
        BACKUP ARCHIVELOG ALL DELETE ALL INPUT;
```

Fast Recovery Area Space Usage (continued)

If you use a backup solution other than RMAN, you still have to use RMAN to remove the files from the Fast Recovery Area. After the archived redo log files have been backed up and removed from disk, use the RMAN `CROSSCHECK` and `DELETE` commands to reclaim the archived log space from the Fast Recovery Area. You should do this on a regular basis or after every backup.

You can also use the Manage Backups page of Enterprise Manager to manage backups. On that page, you can perform a cross-check operation and delete expired and obsolete backups.

Configuring an Archived Redo Log Deletion Policy

You can use the `CONFIGURE ARCHIVELOG DELETION POLICY` command to specify when archived redo logs are eligible for deletion. This deletion policy applies to *all* archiving destinations, including the FRA.

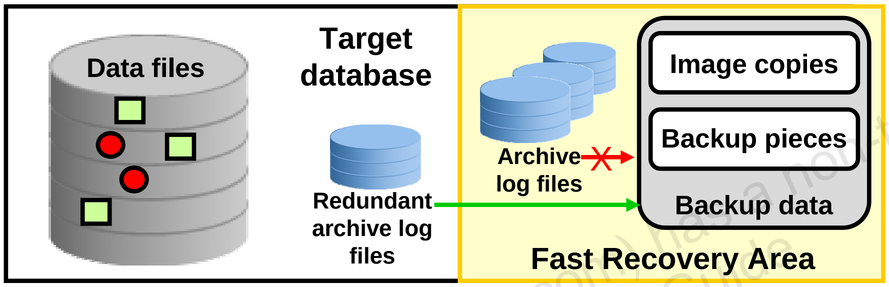
Archived redo logs can be deleted automatically by the database or as a result of user-initiated RMAN commands:

- Only logs in the FRA can be deleted automatically by the database. For archived redo log files in the FRA, the database retains them as long as possible and automatically deletes eligible logs when additional disk space is required.
- You can manually delete eligible logs from any location, whether inside or outside the FRA, when you issue `BACKUP ... DELETE INPUT` or `DELETE ARCHIVELOG`.
- The default is:

```
  CONFIGURE ARCHIVELOG DELETION POLICY TO NONE;
```

What Is Done Automatically for You

- Simplified archive log management in a multiple-component environment
- Increased availability by failover of backup to optional destinations



What Is Done Automatically for You

Simplified Archive Log Management in a Multiple-Component Environment

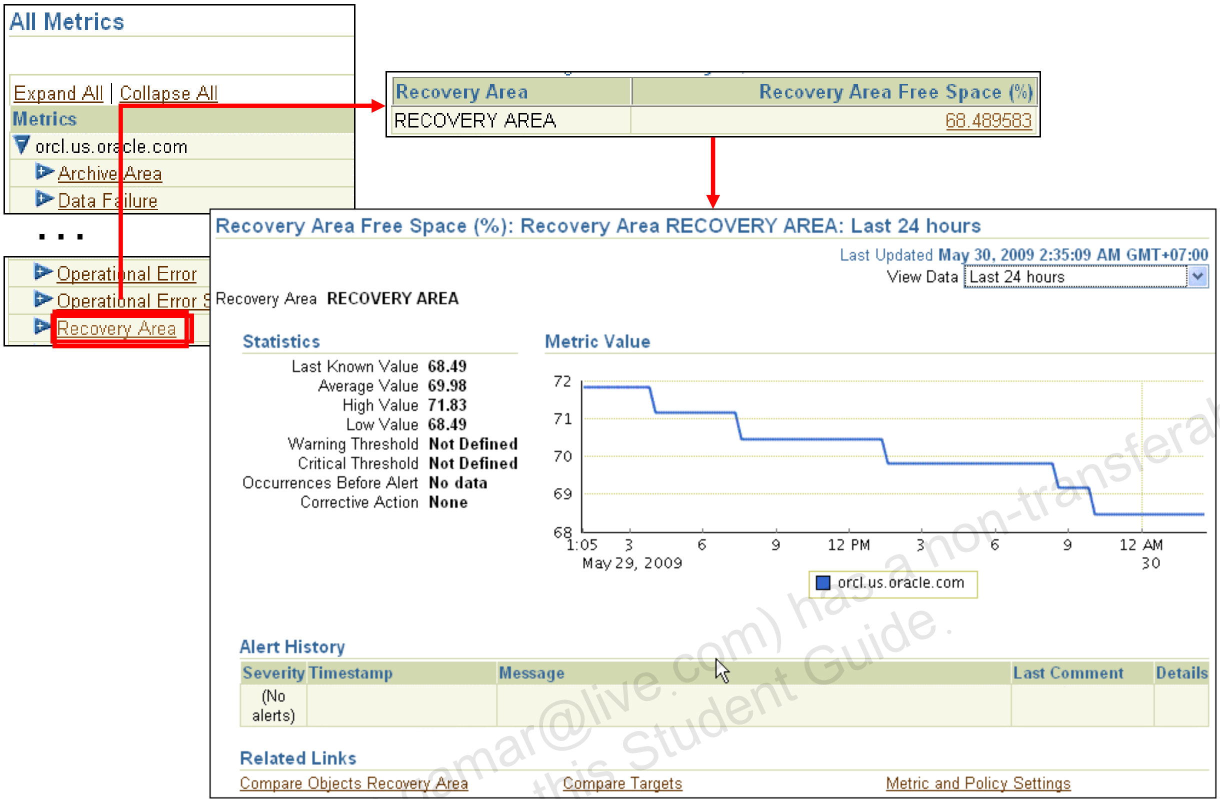
This feature simplifies archive log management when used by multiple components, such as Streams. It also increases availability when backing up archive logs, when an archive log in the FRA is missing or inaccessible.

Enhanced Configuration of Deletion Policies

Archived redo logs are eligible for deletion only when not needed by any required components such as Data Guard, Streams, Flashback Database, and so on. When you configure an archived log deletion policy, the configuration applies to all archiving destinations, including the FRA. Both `BACKUP` ... `DELETE INPUT` and `DELETE`... `ARCHIVELOG` use this configuration, as does the FRA.

When you back up the recovery area, RMAN can fail over to other archived redo log destinations if the archived redo log in the FRA is inaccessible or corrupted.

Monitoring the FRA



All Metrics

Expand All | Collapse All

Metrics
- orcl.us.oracle.com
  - Archive Area
  - Data Failure
  - . . .
  - Operational Error
  - Operational Error S
  - Recovery Area

Recovery Area	Recovery Area Free Space (%)
RECOVERY AREA	68.489583

Recovery Area Free Space (%): Recovery Area RECOVERY AREA: Last 24 hours

Last Updated May 30, 2009 2:35:09 AM GMT+07:00

View Data Last 24 hours

Recovery Area RECOVERY AREA

Statistics

Last Known Value	68.49
Average Value	69.98
High Value	71.83
Low Value	68.49
Warning Threshold	Not Defined
Critical Threshold	Not Defined
Occurrences Before Alert	No data
Corrective Action	None

Metric Value

Alert History

Severity	Timestamp	Message	Last Comment	Details
(No alerts)				

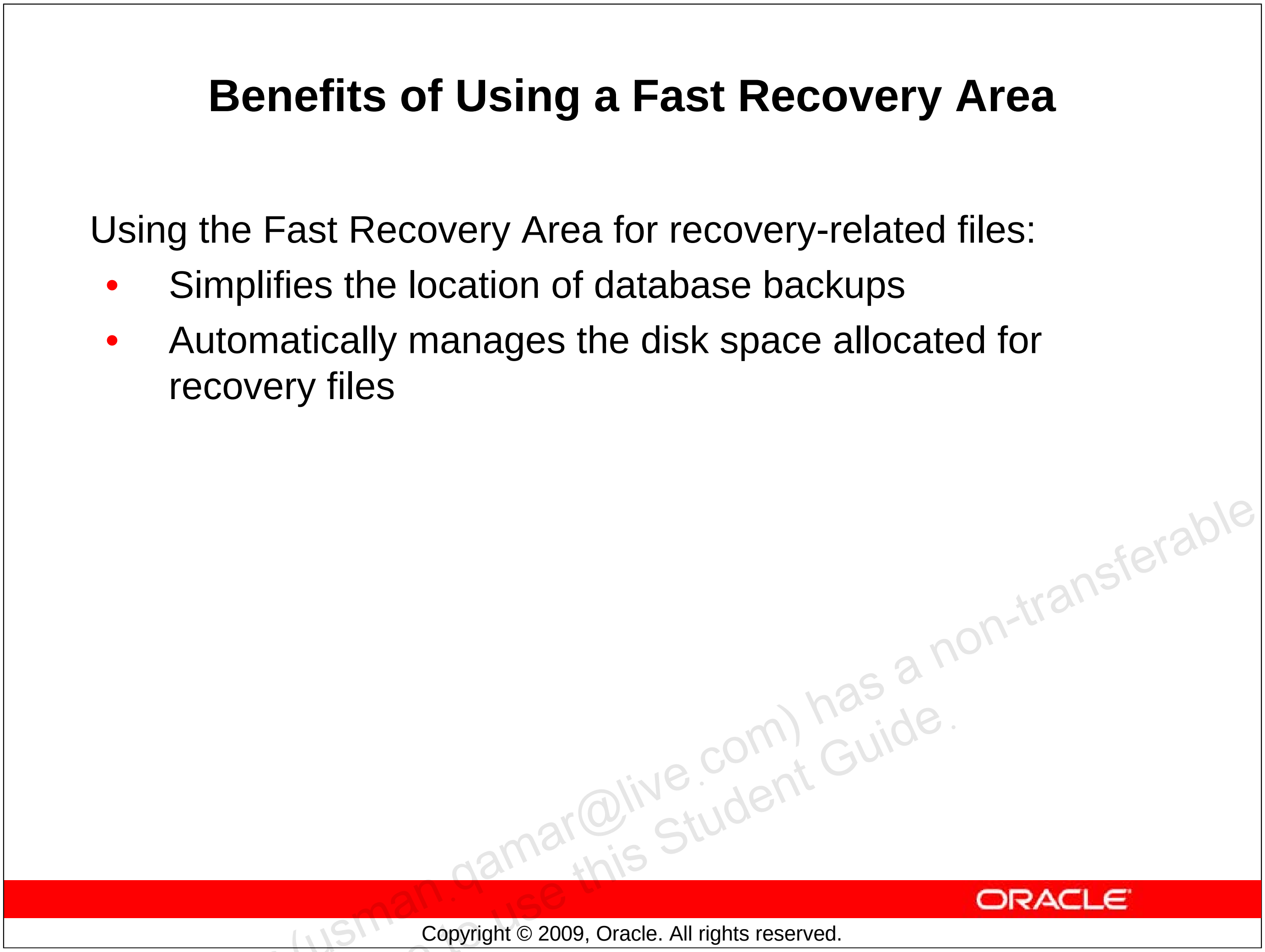
Related Links

Compare Objects Recovery Area | Compare Targets | Metric and Policy Settings

Monitoring the FRA

Real-time FRA metrics can be viewed through Enterprise Manager Database Control. On the Home page, scroll down to the Related Links section and select All Metrics. Scan the list and click Recovery Area.

The displayed page shows the Recovery Area Free Space (%) metric, which indicates the percentage of the recovery that is free space. Click the percentage number to see the graph of recovery area usage.



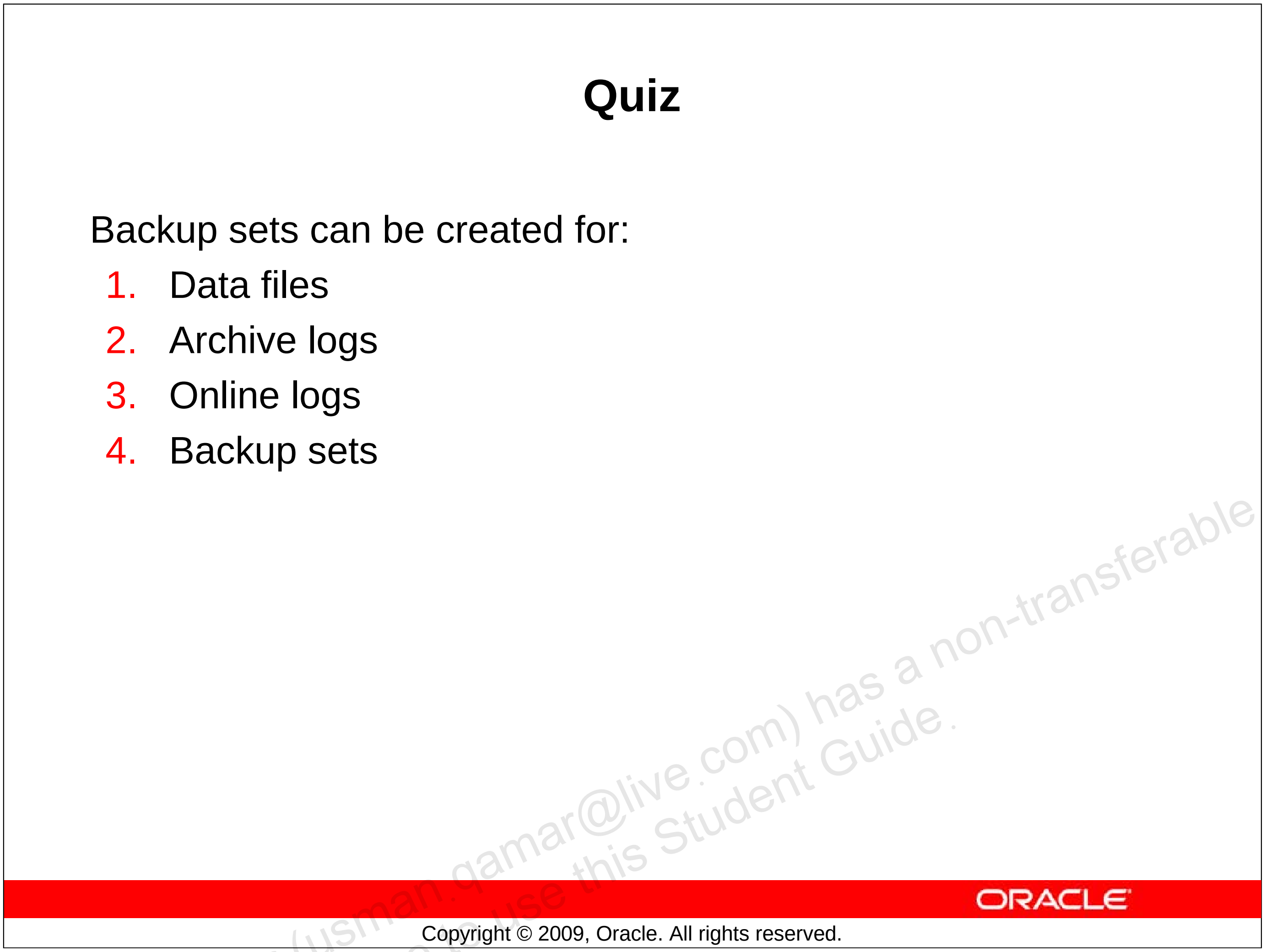
Benefits of Using a Fast Recovery Area

Using the Fast Recovery Area for recovery-related files:

- Simplifies the location of database backups
- Automatically manages the disk space allocated for recovery files

Benefits of Using a Fast Recovery Area

Using a FRA for all recovery-related files simplifies the ongoing administration of your database. Oracle Corporation recommends the use of the Fast Recovery Area for all recovery-related files.

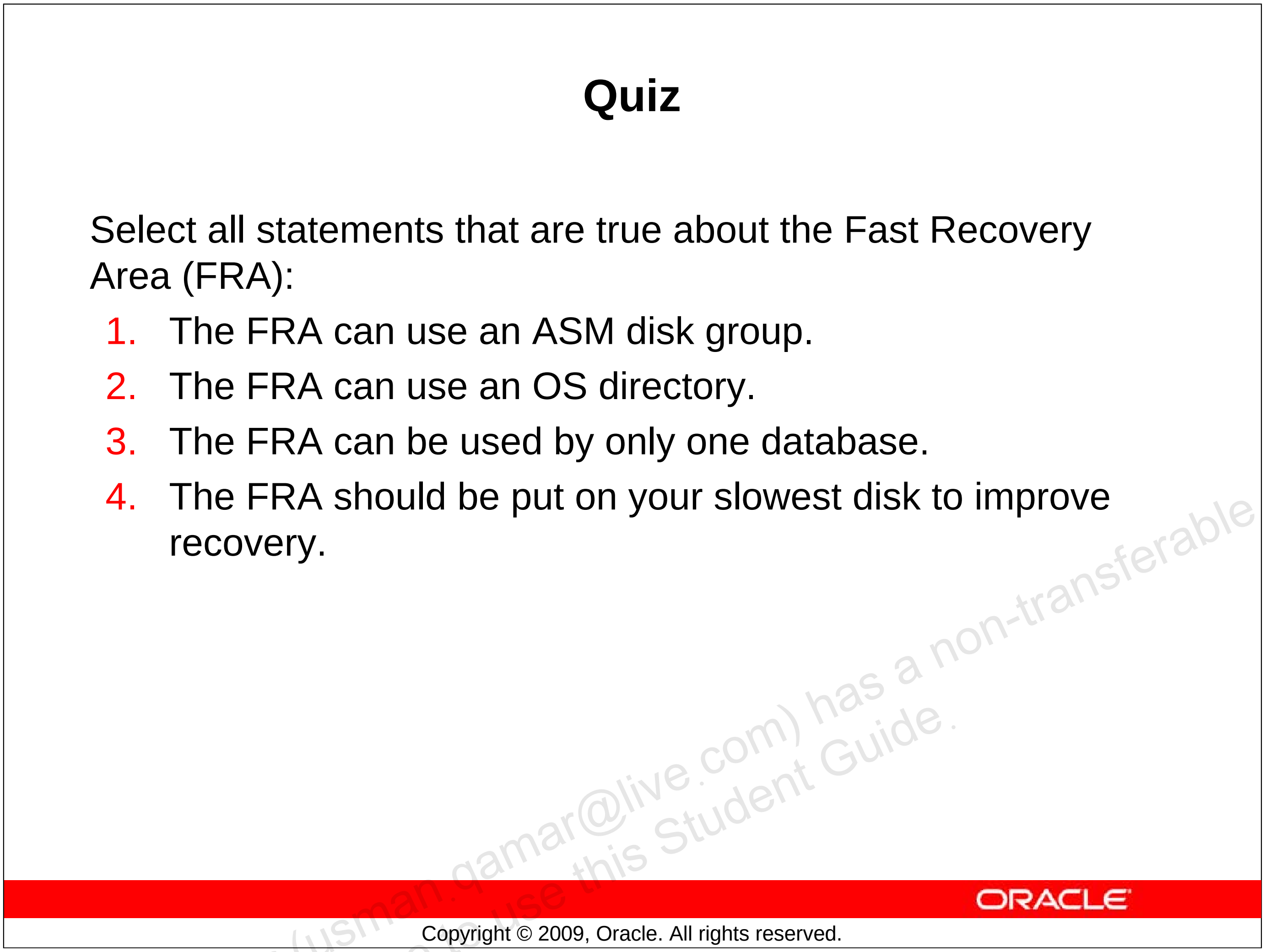


Quiz

Backup sets can be created for:

1. Data files
2. Archive logs
3. Online logs
4. Backup sets

Answer: 1, 2, 4



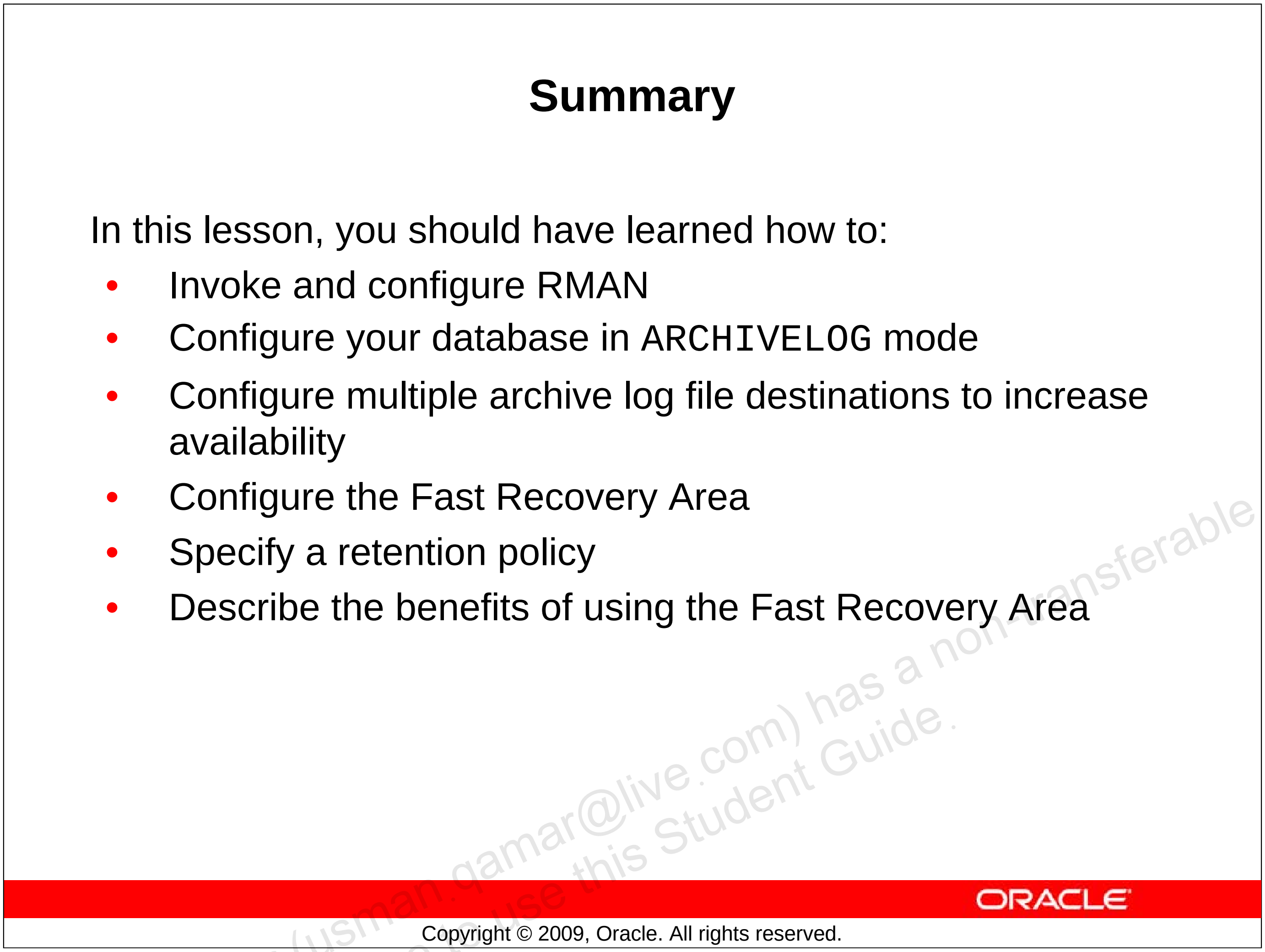
Quiz

Select all statements that are true about the Fast Recovery Area (FRA):

1. The FRA can use an ASM disk group.
2. The FRA can use an OS directory.
3. The FRA can be used by only one database.
4. The FRA should be put on your slowest disk to improve recovery.

ORACLE

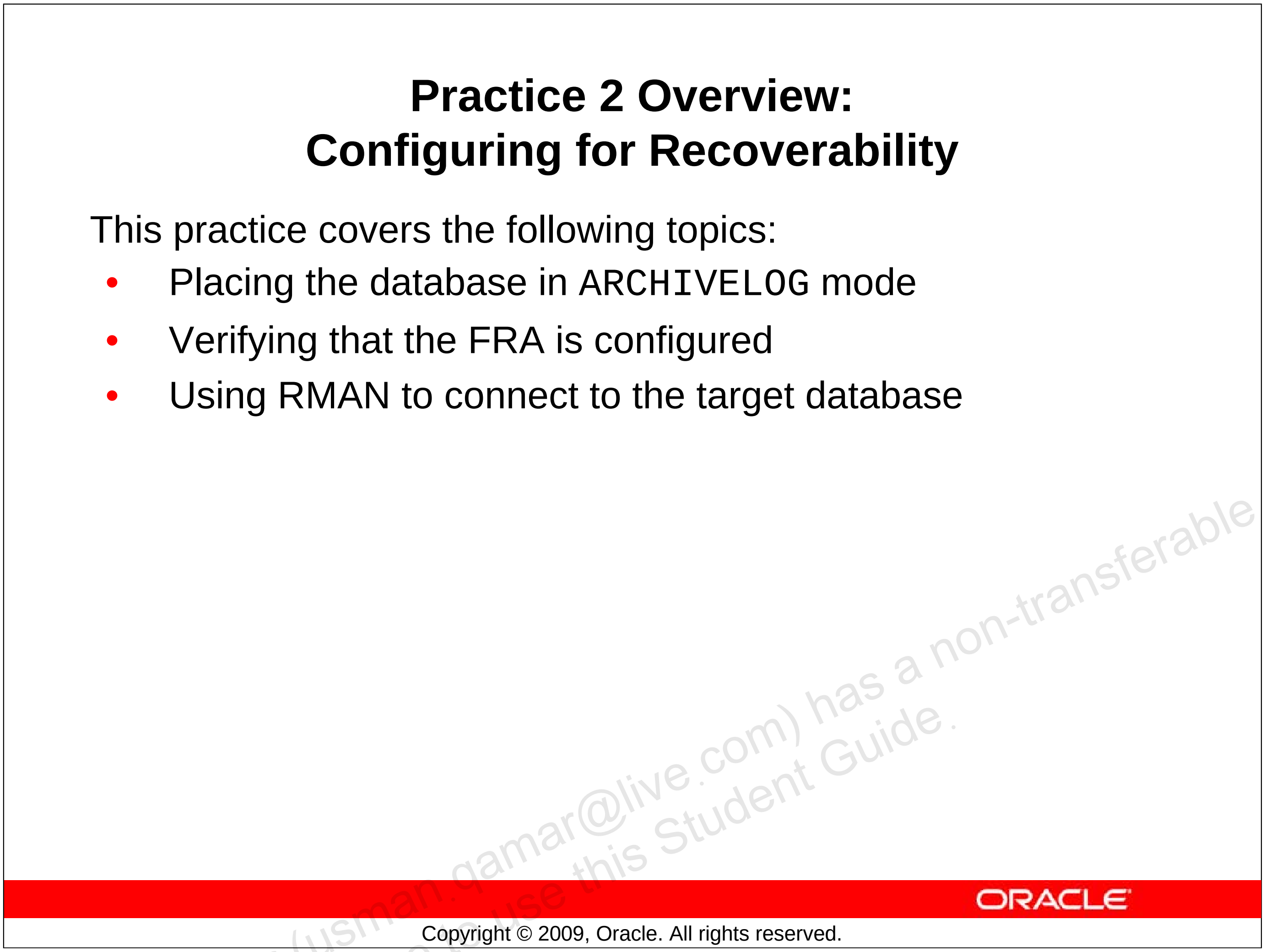
Answer: 1, 2



Summary

In this lesson, you should have learned how to:

- Invoke and configure RMAN
- Configure your database in `ARCHIVELOG` mode
- Configure multiple archive log file destinations to increase availability
- Configure the Fast Recovery Area
- Specify a retention policy
- Describe the benefits of using the Fast Recovery Area



Practice 2 Overview: Configuring for Recoverability

This practice covers the following topics:

- Placing the database in `ARCHIVELOG` mode
- Verifying that the FRA is configured
- Using RMAN to connect to the target database