10 Automatic Storage Management

CERTIFICATION OBJECTIVES

Automatic Storage Management (ASM) is one of Oracle Database 10g's most important and revolutionary enhancements. The Oracle Managed Files (OMF) feature, which has Oracle create and name files for you, was introduced a couple of versions ago. ASM is Oracle's new feature that lets you finally completely automate your file management tasks. ASM even enables you to bypass using a third-party Logical Volume Manager, mirroring and striping your disks directly from within Oracle.

This chapter introduces you to ASM by first exploring its architecture. The chapter then takes you through the details of administering ASM in your database. Administering ASM means managing the ASM instance, disk groups, and ASM files. You'll also learn how to migrate to an ASM system using the RMAN.

Let's start with a review of the nature of ASM and its architecture.

CERTIFICATION OBJECTIVE 10.01

Introduction to Automatic Storage Management

Oracle DBAs sometimes maintain hundreds or even thousands of datafiles for each of the databases they manage. File and I/O management, in fact, is typically one of the largest consumers of a DBA's time. Oracle introduces the new Automatic Storage Management (ASM) system to simplify your management tasks by automating disk and file management.

Every bit of data you have in Oracle is stored somewhere on a disk. Your job is to help retrieve that data from disk as quickly as possible. Disk I/O is much slower than accessing data currently in random access memory (RAM), and it often turns out to be the single biggest bottleneck for busy OLTP systems. ASM not only simplifies database storage administration, but it also optimizes the database storage layout for the best performance. ASM is built on top of OMF, and it takes OMF much further than simply having Oracle create and name files for you. ASM acts as Oracle's own Logical Volume Manager (LVM), by handling striping and mirroring functions previously done by third-party tools. DBAs spend a lot of time worrying about I/O load balancing. ASM has features that automatically perform load balancing for you!

exam watch

You can't use operating system commands or utilities to access ASM files. You must use the RMAN to copy ASM files.

You don't need to move to the new ASM system overnight. Instead, you can continue using your present file systems and slowly migrate to ASM files over time. Oracle lets you mingle operating system, OMF, and ASM files together in a single database.

Benefits of ASM

ASM enables you to manage data by selecting the desired reliability and performance characteristics for classes of data, rather than interacting with large storage systems on a per-file basis. ASM file systems offer several benefits, including the following:

- ASM prevents disk fragmentation, so there won't be any need to perform time-consuming relocation of data.
- You can keep all your datafiles and tablespaces in an ASM storage system. ASM will manage your datafiles, control files, redo logs, archive logs, RMAN backup sets, and so on. You can manage database objects such as tablespaces without needing to specify and track filenames.
- Your DBA file management duties are simpler, because you'll be dealing with a few disk groups, instead of directly handling numerous datafiles. Managing very large databases becomes simple when you use an ASM system.
- ASM performs mirroring and striping, thus increasing reliability and performance. Mirroring is applied on a file basis, rather than on a disk basis, thus giving you more control over which files you want to protect.
- ASM automatically balances I/O load in parallel across all available disk drives to prevent hot spots and maximize performance. When you add new disks, ASM automatically moves data around to balance I/O load among the disks. ASM load-balances file activity by uniformly distributing file extents across all disks in a disk group. This automatic online disk space reorganization saves you plenty of time.
- ASM helps you maintain redundant copies of data to provide fault tolerance. If you wish, you can build an ASM storage system on top of vendor-supplied reliable storage mechanisms.
- ASM is free! You don't pay anything extra for the ASM feature, since it's part of your Oracle Database 10*g* Server software.

exam watch ***You can use an ASM storage system for any Oracle database files, including datafiles, control files, online redo log files, archived redo logs, and RMAN backup sets.***

ASM Architecture

ASM has three important components: the ASM instance, disk groups, and ASM files. ASM runs as a tiny enterprise of its own, with its own instance and background processes, helping Oracle Database 10*g* manage its files.

ASM files are at the core of all ASM activity. You use ASM so you can use its files for your database. Unlike in the case of normal Oracle operating system-based database files, you don't directly access ASM files. ASM files are part of larger entities called *ASM disk groups*, which act as the default location for all your database files. You use ASM disk groups to get to the ASM files. When you use ASM files, you don't need to refer to your tablespaces by filenames; you use simple disk group names instead.

You don't need to change the way you manage your Oracle database when you switch to an ASM storage system. You can even have your current operating system-based files coexist with new datafiles that you create as ASM files. All your logical concepts like extents, segments, and tablespaces remain intact in an ASM system.

Here's a summary of the basic characteristics of an ASM-based storage system:

- You can store all Oracle database files as ASM files. There is a one-to-one mapping between an Oracle database file (datafile, control file, and so on) and an ASM file.
- An ASM disk group consists of a set of disk drives.
- A database can have multiple disk groups.
- An ASM disk group can also contain files from several disk groups.
- An ASM file is always spread over *all the disks* in an ASM disk group. An ASM file can belong to only one disk group.
- ASM allocates disk space in units called *allocation units*. All ASM disks are partitioned in allocation units of 1MB.

In order to use an ASM file system, you must first start an ASM instance, which is somewhat like a small database instance (without its own database files). The ASM instance manages the disk groups, and its main purpose is to help the database access the ASM files. The Oracle database contacts the ASM instance for information about the ASM datafiles, and then accesses those files directly on its own. Let's look at the ASM instance in detail in the following section.

CERTIFICATION OBJECTIVE 10.02

Managing the ASM Instance

Before you can use the ASM files, you must ensure that the ASM instance is up and running. You create an ASM instance just as you would create any other Oracle instance. The big difference is that you have only a handful of initialization parameters to set up for an ASM instance. In addition, an ASM instance doesn't mount any Oracle database files. Its main job is to maintain ASM file metadata, so the database can use it to access the files directly.

on the job

In order to use ASM, you must have an ASM instance running on your server.

Unlike a normal Oracle database,an ASM instance doesn't have a data dictionary, making it necessary for you to connect only as an administrator, either through operating system authentication as SYSDBA or SYSOPER, or by using a password file, if you're connecting remotely.

Here's a summary of the ASM instance's functions:

- Manage disk groups
- Protect disk groups
- Communicate file metadata to database instances using the ASM files

Creating an Instance

To create an ASM instance, you must have the SYSDBA privilege. You can perform most managing tasks (except creating the instance and a few others) with just the SYSOPER privilege. Of course, you must also belong to an operating system group that has the privileges to connect as a SYSDBA, a group like the typical dba group. If you connect as a SYSDBA, you'll have complete administrative privileges in an ASM instance. If you connect as a user with the SYSOPER privilege, you'll have the ability to execute the following commands:

- `STARTUP/SHUTDOWN`
- `ALTER DISKGROUP CHECK`
- `ALTER DISKGROUP MOUNT/DISMOUNT /REMOUNT`
- `ALTER DISKGROUP OFFLINE`

exam watch

You must issue all disk group management commands from within an ASM instance. You database has no direct connection to ASM disk groups.

In addition to the right to execute these commands, a SYSOPER privilege holder can also use all the data dictionary views associated with an ASM instance. The SYSDBA privilege lets you perform all the listed commands, as well as the more powerful CREATE DISKGROUP, ADD DISK, DROP DISK commands and the RESIZE clause in the ALTER DISKGROUP statement..

on the job

An ASM instance takes up about 100MB of space. Most ASM instances should need no more than 64MB of SGA.

Initialization Parameters for the ASM Instance

You must create an initialization parameter file to create a new ASM instance. Fortunately, since the ASM instance doesn't have its own datafiles and the like, you'll need to configure only a minimal number of initialization parameters.

on the job

If you set only one parameter—`INSTANCE_TYPE`—correctly (`INSTANCE_TYPE=ASM`), Oracle will start up the ASM instance with default values for all the other parameters.

The following are the key initialization parameters that you must configure for your ASM instance.

exam watch

If you don't explicitly set the `INSTANCE_TYPE` parameter to `ASM`, your ASM instance will fail to start. The default value for this parameter is `RDBMS`, which is applicable to normal Oracle databases.

- **INSTANCE_TYPE** You must set the INSTANCE_TYPE parameter to ASM.
- **DB_UNIQUE_NAME** You normally won't need to set this parameter, as it applies only to ASM within a cluster or on a node. The parameter shows the unique name for a group of ASM instances in a cluster or on a node. The default value for this parameter is +ASM. Change this only if you think you're going to have more than one ASM instance running on the same node.
- **ASM_POWER_LIMIT** This parameter indicates the maximum speed to be used by this ASM instance during a disk *rebalance* operation. When you add or delete individual disks, ASM moves around an amount of data equal to the storage you are adding or reducing from a disk group. ASM does this so it can evenly redistribute the datafiles and balance I/O load across the disks. The default for this parameter is 1, and the range is 1 to 11 (1 is slowest and 11 fastest). You have the option of specifying the rebalance speed by using the POWER clause in the disk REBALANCE command, as in this example:

```
ALTER DISKGROUP dgroup1 REBALANCE POWER 4;
```

exam watch

If you leave the `ASM_DISKSTRING` parameter set as NULL, ASM will find all disks to which it has read/write access during a disk-discovery process.

- **`ASM_DISKSTRING`** This parameter sets the disk location for Oracle to consider during a disk-discovery process. When you add a new disk to a disk group, for example, the ASM instance will discover the new disk by looking up the directories listed in its `ASM_DISKSTRING` parameter. Thus, by setting this parameter, you can limit the number of disks that ASM considers for disk discovery. The default for this parameter is NULL. It can take a list of values, as shown here:

```
ASM_DISKSTRING ='/dev/rdsk/*s1', '/dev/rdsk/c1*
```

- **`ASM_DISKGROUPS`** This parameter lets you specify the name of any disk group that you want the ASM instance to automatically mount at instance startup. The default value for this parameter is NULL. If you use an init.ora text file, you must make sure to add the names of any disk groups that you want to mount when the instance starts up. On the other hand, if you use an SPFILE, Oracle will automatically make the necessary additions and deletions to the SPFILE when you create, add, or drop a disk group.

on the job

The ASM instance uses the `LARGE_POOL` memory buffer. You should allocate at least 8MB to this parameter, so it can serve the ASM instance effectively.

Creating the ASM Instance

You can either create the ASM instance manually, or use the DBCA to create it. You can create an ASM instance through an init.ora file, just as you can create database instances (see Exercise 10-1). An ASM instance usually requires bout 100MB of disk space.

EXERCISE 10-1

Creating an ASM instance

Let's create an init.ora file with the five key ASM-related initialization parameters. Here's our init.ora file:

```
INSTANCE_TYPE=ASM
DB_UNIQUE_NAME= +ASM
ASM_POWER_LIMIT =1
ASM_DISKSTRING = '/dev/rdsk/*s1', /dev/rdsk/c1*'
```

```
ASM_DISKGROUPS = dgroupA, dgroupB
LARGE_POOL_SIZE = 16M
```

Now, export your new ASM instance name and run the following commands:

```
$ export ORACLE_SID=ASM
$ sqlplus /nolog
SQL> connect / AS sysdba
Connected to an idle instance.
SQL> startup
ASM instance started
Total System Global Area 147936196 bytes
Fixed Size 324548 bytes
Variable Size 96468992 bytes
Database Buffers 50331648 bytes
Redo Buffers 811008 bytes
ASM diskgroups mounted
```

Using the Database Configuration Assistant (DBCA) makes it very easy to create an ASM instance. When you create a new Oracle database, as shown in Figure 10-1, the DBCA offers you three storage options: file systems, raw devices, and ASM.

If you choose file systems or raw devices for storage, and you decide to use an ASM file system later on, you'll need to separately create your own ASM instance. However, if you choose ASM for storage, the DBCA will check to see if an ASM instance aready exists on your server. If it does, the DBCA will then show you the disk groups being managed by that ASM instance and ask you to choose the disk groups for your new Oracle database.

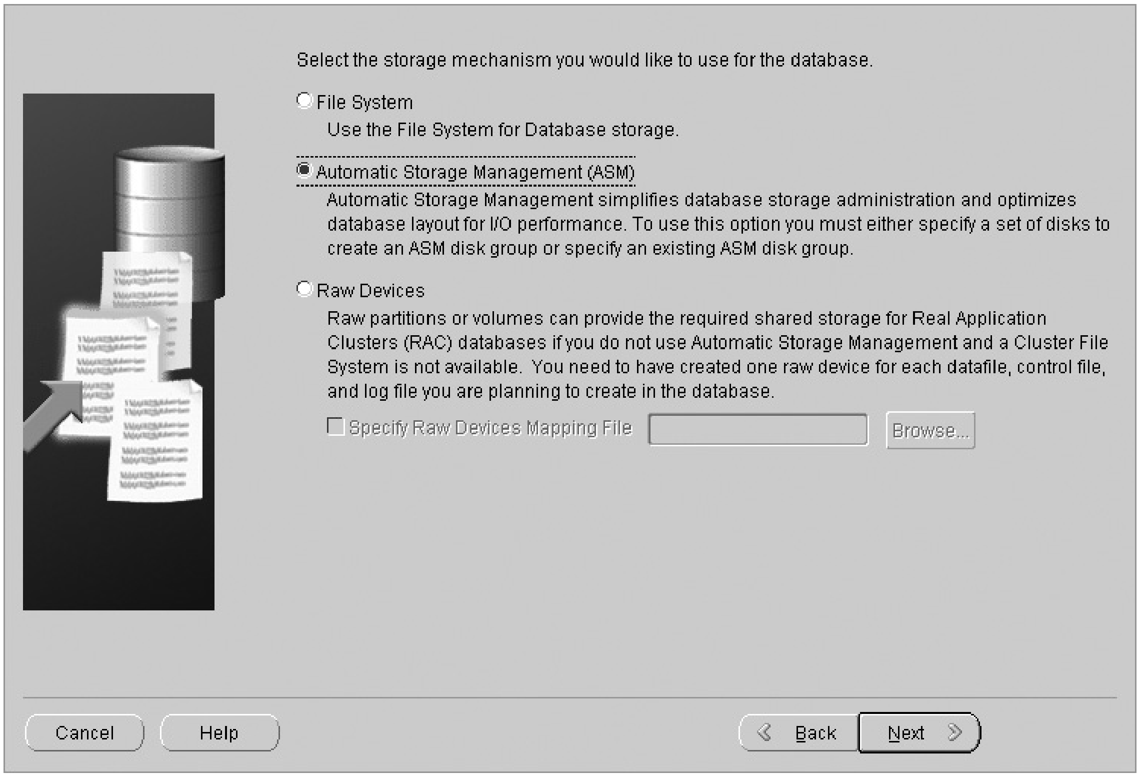
If you choose ASM as your storage mechanism and you haven't already configured an ASM instance, the DBCA will automatically create one for you, after asking you to provide a username and password for the separate SYS user for the ASM instance (you'll need this for remote database access).

The DBCA automatically creates an entry in the oratab file on UNIX systems, so the operating sytem is aware of the new instance. On Windows systems, the DBCA creates the Oracle service and makes the appropriate Windows Registry entries. The DBCA also creates a parameter file and a password file for the new ASM instance.

ASM Instance Architecture

An ASM instance has several background processes like the SMON, PMON, and LGWR processes. In addition, there are two new background processes: ASM Rebalance

FIGURE 10-1 Using the DBCA to configure ASM during database creation



Select the storage mechanism you would like to use for the database.

( ) File System
Use the File System for Database storage.

[✓ Automatic Storage Management (ASM)]
Automatic Storage Management simplifies database storage administration and optimizes database layout for I/O performance. To use this option you must either specify a set of disks to create an ASM disk group or specify an existing ASM disk group.

( ) Raw Devices
Raw partitions or volumes can provide the required shared storage for Real Application Clusters (RAC) databases if you do not use Automatic Storage Management and a Cluster File System is not available. You need to have created one raw device for each datafile, control file, and log file you are planning to create in the database.

☐ Specify Raw Devices Mapping File

Master (RBAL) and ASM Rebalance (ARB*n*). Here's what the two background processes do:

- The RBAL process is in charge of coordinating disk activity.
- The ARB*n* processes (there could be several of these, like ARB0, ARB1, and so on) perform the actual rebalancing work like moving the data extents around.

In addition to the ASM instance background processes RBAL and ARBO, any Oracle database instance that uses an ASM instance will have two new ASM-related background processes, the RBAL and the ASM Background (ASMB) processes. This is what these two new database processes do:

- The RBAL process performs global opens of the disks in the ASM disk groups.
- The ASMB background process connects as a foreground process into your ASM instance. Remember that the ASM instance must be in operation in order for

exam watch

The ASM Rebalance (ARBn) process performs the actual disk rebalancing work.

your database to access ASM files. The ASMB process acts as the link between the ASM instance and your database instance, communicating information like datafile creation, deletion, updating statistics, and performing instance health checks.

Managing an ASM Instance

You can use the OEM Database Control to manage all aspects of the ASM instance. The home page of the Database Control shows the status of your ASM instance. From the ASM home page, you can click the Configuration tab to go the ASM Configuration page, where, you can modify your current ASM instance parameters. You can also go to the Automatic Storage Management home page and check on ASM instance performance issues, like I/O response time and throughput for the disk groups being managed by your ASM instance.

You can also use manual commands to start up and shut down the ASM instance. Let's review the startup and shutdown procedures for ASM instances in the following sections.

exam watch

Oracle databases expect the ASM instance to be always running, so they can access the ASM files. You should set up automatic startup procedures for ASM instances following a host bootup. In Windows, for example, you should make the ASM instance an automatically started service.

Starting an ASM Instance

The `STARTUP` command for an ASM instance is quite similar to the `STARTUP` command for your Oracle databases, with a couple of interesting differences. Make sure you set the `INSTANCE_TYPE` parameter to `ASM`, so Oracle knows it is an ASM instance, not a regular Oracle database.

exam watch

When starting an ASM instance, you can use the `STARTUP` command with the `NOMOUNT`, `MOUNT`, `RESTRICT` and `FORCE` options. You can't use the `STARTUP OPEN` syntax.

During the mount phase of the normal Oracle `STARTUP` command, an Oracle database reads the control file and mounts the file systems specified in the control file. An ASM instance doesn't have any file systems to mount. Rather, an ASM instance mounts the disk groups specified by the initialization parameter ASM_DISKGROUPS. The NOMOUNT command works in an analogous way to the way it works in regular Oracle databases: it starts the ASM instance without mounting any disk groups.

exam watch

In order to use the ASM feature, you must have your ASM instance running. This, however, means that you just start the ASM instance in the MOUNT mode. There are no datafiles to open in an ASM instance, so you don't use the OPEN option for the STARTUP command.

When you issue a STARTUP FORCE command, the ASM instance is shut down with a STARTUP ABORT command before restarting it. If you use the STARTUP RESTRICT command, it prevents any client Oracle database instances from connecting to the ASM instance.

exam watch

If you either start up your ASM instance with the STARTUP RESTRICT command or issue the ALTER SYSTEM ENABLE RESTRICTED SESSION command in a normal ASM instance, Oracle database instances can't connect to the ASM instance.

Shutting Down an Instance

You shut down an ASM instance just as you would shut down a normal Oracle database. Here's an example:

```
$ sqlplus /nolog
SQL> connect / AS sysdba
Connected.
SQL> shutdown normalOn the Job: If the ASM instance fails, the
database instances connected to it will also shut down. However,
the reverse isn't true; the failure of a database instance has
no impact on an ASM instance.
```

There are some caveats, however, in using the SHUTDOWN command in the case of an ASM instance. The up and down status of all Oracle databases connected to an ASM instance critically depend on the ASM instance's status. If you shut down an ASM instance, all Oracle databases currently connected to it will also shut down. When you issue a SHUTDOWN command to an ASM instance, it forwards the SHUTDOWN command, in the same mode, to the Oracle databases that are connected to the ASM instance.

exam watch

When an Oracle database uses ASM files by connecting to an ASM instance, it will stay open only as long as its connection to the ASM instance is intact. If the connection terminates, the Oracle instance will terminate as well. Any `SHUTDOWN` command you use in the ASM instance will also apply, in the same mode, to all connected databases. For example, if you shut down the ASM instance with | the `SHUTDOWN IMMEDIATE` comamnd, all databases connected to that instance will also shut down in the `SHUTDOWN IMMEDIATE` mode.

If you shut down an ASM instance in the NORMAL mode (use the SHUTDOWN NORMAL command or just the SHUTDOWN command), the ASM instance waits for all connected Oracle database instances to terminate their ASM connections before shutting down. In the IMMEDIATE (and TRANSACTIONAL) mode, an ASM instance waits until all currently executing SQL in the dependent databases completes, but doesn't wait for the database instances to disconnect.

If you issue the SHUTDOWN ABORT command, the following events occur:

- The ASM instance instantly aborts.
- All open connections to the ASM instances are terminated.
- As a result of connections to the ASM terminating, all dependent Oracle databases will terminate immediately.

CERTIFICATION OBJECTIVE 10.03

Managing ASM Disk Groups

An ASM *disk group* is a collection of disks that is somewhat analogous to the logical volumes created by an LVM from underlying physical disks. You manage the underlying disks of a disk group indirectly, by managing the disk group. Thus, even if you have a large number of disks, you can aggregate them into a very small number of disk groups, which makes life easy for you. When you add storage to your ASM system, you simply add disks to an ASM disk group. Therefore, if your database is growing at a fast rate, the total stoarge will increase, but the number of disk groups could remain at a small, stable number.

Providing Performance and Redundancy with Disk Groups

One of the biggest selling points in switching to ASM file management for Oracle databases is that it offers you both additional performance and protection, while decreasing the management overhead, especially for large, mission-critical databases. These benefits are similar to the benefits offered by LVM tools provided by third-party vendors. However, the big advantage of ASM over the third-party tools is that the Oracle DBA can take over most of the disk management tasks when using an ASM-based storage system.

You don't need to be an expert in file sytems, RAID, or logical volumes to use ASM files in your database. If you understand how ASM manages disk groups and how Oracle accesses the database files that are spread over the the ASM disks, you are ready to use ASM.

ASM can perform both striping and mirroring tasks better than a third-party LVM, because it understands the Oracle file types and uses an appropriate strategy for each type.

ASM provides both performance and redundancy, the first through *striping* and the second through the *mirroring* of data on the disk drives. Let's look at these two features in the following sections.

ASM Striping

ASM systems use disk groups and disks, with your database files being stored on the ASM disks. The way you place (or write) your database files over ASM disks plays a critical role in determining I/O performance. To provide you with optimal I/O performance, ASM *stripes* files across all the disks that are part of a disk group. For performance reasons, you must use disks of the same type and performance capacity in a disk group.

ASM provides two types of data striping, depending on the database file type:

- **Coarse striping** The most common striping scheme in an ASM system is *coarse* striping, so called because the stripe size is a relatively large 1MB chunk of file space. You may use coarse striping for all files in an Oracle database, except the control files, online redo log files, and flashback files.
- **Fine striping** Some of your database files, like redo log files, may need faster access than others. To reduce file latency, ASM provides a *fine* striping scheme, where the striping is in smaller chunk sizes of 128KB. If you have 20 disks in your disk group, ASM would stripe your redo logs in 128KB sized chunks across

all 20 of the disks. Thus, when you perform I/O operations, you can access your file data in parallel, thus improving performance. You may want to use fine striping for control files, online redo log files, and flashback files.

ASM Mirroring

Disk mirroring provides data *redundancy*. If you lose a disk, you can use its mirror disk to continue operations without missing a beat. ASM mirroring is not the same as an operating sytem-level mirroring scheme, although the goal of both is to provide redundancy. Operating system-based LVMs mirror entire disks. ASM mirrors *extents*. Whenever ASM allocates an extent (also called the *primary extent*, as opposed to a mirrored extent), it simultaneously allocates a mirror copy of the extent to one of the disks in the same disk group. A disk could have its mirror extents on one or several disks in the disk group.

When any disk in a disk group fails, ASM reconstructs the failed disk on the fly by using the mirrored extents from the other disks in the disk group. What's the advantage to mirroring extents rather than mirroring disks? When ASM is reconstructing a lost disk, your storage system will take a smaller I/O hit, because several disks are sharing the I/O necessary to reconstruct the failed disk device.

Failure Groups You can lose the services of a disk not only when the disk drive fails, but also if a shared resource such as a disk controller bites the dust. When a SCSI disk controller fails, all the disks connected to it will be inaccessible. A set of disks that can all fail because they share a common resource, is called a *failure group*. When you are trying to protect your data by introducing redundancy, you do so on the basis of failure groups. That is, if you want to ensure that you have redundancy, you must store your mirrored copy in a separate failure group.

ASM never places a primary extent and its mirror copy in the same failure group. Thus, even if you lose several disks in a failure group, ASM can survive the disaster, by starting the reconstruction of the lost devices from the mirrored copies of their extents from disks that are outside the disk's failure group.

Types of ASM Mirroring ASM supports three types of disk mirroring, each providing a different level of data redundancy. When you create a new ASM disk group, you need to decide the level of disk mirroring you need:

- **External redundancy** This type really doesn't provide any mirroring. You choose this level of mirroring when you are using operating system storage array protection. Disk groups under this redundancy level don't have any failure groups.

- **Normal redundancy** This type provides two-way mirroring. Since you mirror through the creation of failure groups, you must have *two failure groups*, each group relying on a separate disk controller. Two-way mirroring means that when one copy of a file extent is written to a disk in failure group Group A, a mirrored copy of the file extent is simultaneously written to a disk in failure group Group B. Thus, to support a normal redundancy level, you must create at least *two* failure groups.
- **High redundancy** This type provides three-way mirroring, which means you should have *three failure groups*, each controlled by a separate disk controller. When one copy of a file extent is written to a disk in failure group Group A, a mirrored copy of the file extent is simultaneously written to both a disk in Group B and a disk in Group C. Thus, to support a normal redundancy level, you must create at least *three* failure groups.

Creating a Disk Group

The easiest way to create a disk group is to use the Databae Control's Disk Group Adminstration page. You can select the redundancy level, disk group name, and the list of disks that you want to be part of a disk group.

You can also create a disk group manually by using the `CREATE DISKGROUP` command. For example, suppose that you have three SCSI disk controllers and a total of twelve disks. Disks Diska1 through Diska4 are on a separate SCSI controller from disks Diskb1 through Diskb4. Similarly, disks Diskc1 through Diskc4 are on yet another disk controller. You can create three failure groups, each with four disks. The first four disks, Diska1–a4, will be on disk controller 1; the second four disks, Diskb1–b4, will be on disk controller 2; and the last four disks, Diskc1–c4, will be on disk controller 3.

First, start the ASM instance in the `NOMOUNT` mode. (If you want to access previously created diskgroups, you must use the `MOUNT` mode.) If none exists, the instance is ready for you to create a disk group now. Then you can create your three disk groups to correspond with your three failure groups, using the `CREATE DISKGROUP` command, as shown here:

```
% sqlplus /nolog
SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup nomount
SQL> create diskgroup  test_group1 high redundancy 2  failgroup groupA disk
  3 ‘/devices/diska1’,
  4 ‘/devices/diska2’,
  5 ‘/devices/diska3’,
  6 '/devices/diska4',
  7 failgroup groupB disk
  8 '/devices/diskb1',
  9 '/devices/diskb2',
 10 '/devices/diskb3',
 11 '/devices/diskb4';
 12 failgroup groupC disk
 13 ‘/devices/diskc1’,
 14 ‘/devices/diskc2’,
 15 ‘/devices/diskc3’,
 16 ‘/devices/diskc4’,
```

Oracle uses the search string of the format `/devices/diska1` to find the disks on your system. The use of the `FAILGROUP` and `REDUNDANCY` keywords is purely optional. If you don’t specify the `FAILGROUP` keyword, each disk in the disk group will be in its own failure group.

Here are the implications of using the `HIGH REDUNDACY` setting for the new disk group, test_group1:

- There are three failure groups, each defined by the `FAILGROUP` keyword (you must have at least three failure groups if you specify high redundancy).
- Each of the failure groups has four disks.
- When Oracle writes data to the disks in the first failure group, GroupA, it also writes those extents to disks in the other two failure groups, GroupB and GroupC.

exam watch

When you create a new disk group, ASM will automatically mount it. Don’t forget to add the diskgroup’s name to your init.ora text file, if you don’t already have it there when you start the database in the `NOMOUNT` mode. If you put the name of the disk group in the inita.ora file, Oracle will automatically mount it the next time you start the instance. Of course, if you’re using an SPFILE instead, Oracle will automatically write the new disk group’s name in the SPFILE.

EXERCISE 10-2

Create a Disk Group

Create a disk group with two disk groups with normal redundancy. Notice the two failure groups that this level of redundancy implies.

```
SQL> create diskgroup  dgroup1 normal redundancy
       failgroup controller1 disk
       ‘/dev/rdsk/c0t0d0s2’ name testdisk size 100G,
       ‘/dev/rdsk/c0t1d0s2’,
       ‘/dev/rdsk/c0t2d0s2’
       failgroup controller2 disk
       ‘/dev/rdsk/c1t0d0s2’,
       ‘/dev/rdsk/c1t1d0s2’,
       ‘/dev/rdsk/c1t2d0s2’;
```

Adding Disks to a Disk Group

You use the ALTER DISKGROUP command to add a new disk to a disk group, as shown here:

```
SQL> alter diskgroup test_group1 add disk
     '/devices/diska5' name diska5,
     '/devices/diska6' name diska6,
```

There are two interesting points to note in this example:

- There is neither a FAILGROUP nor a REDUNDANCY specification for the two new files. When you don’t specify a failure group for a disk, the disk is in its own failure group.
- There is a new NAME clause in this example. In the previous example, there was no NAME clause. There, Oracle would assign its own system-generated names.

Dropping Disks and Disk Groups

You use the ALTER DISKGROUP command to drop a disk from a disk group, as shown here:

```
SQL> alter diskgroup test_group1 drop disk diska5;
```

You can use the UNDROP clause to keep a pending DROP DISK command from taking place. If you've already dropped the disk, you can't retrieve it with the UNDROP clause. If you've used the optional FORCE clause when you dropped a disk, or if you've issued a DROP DISKGROUP command, you can't use the UNDROP clause. Here's an example demonstrating how to use the UNDROP clause in an ALTER DISKGROUP command:

```
SQL> alter diskgroup test_group1 undrop disks;
```

This command will cancel the pending drop of all disks from the disk group test_group1.

In order to remove an entire disk group, you use the following command, after putting the database in the MOUNT state.

```
SQL> drop diskgroup test_groupA including contents;
```

Rebalancing Disk Groups

One of the perennial concerns for a DBA is the existence of hot spots in the disk system, which may lead to I/O bottlenecks. ASM rebalances a disk group automatically and dynamically, whenever you add or remove a disk from a disk group. ASM strives for a constant I/O balance across all the disks in a disk group. Thus, when you add new disks or remove some disks, this I/O balance is disturbed, but ASM sets it right automatically. It does this by simply moving just enough data around to match the space you added or removed. For example, if you add a 14GB size disk, ASM will reassign only 14GB of datafiles to rebalance the disk group.

Since there is some I/O impact on your system during disk group rebalancing operations, you may be better off consolidating your disk add and remove operations, so ASM needs to perform fewer rebalancing operations. Here's an example of the disk rebalancing command:

```
SQL> alter diskgroup dgroup1 rebalance power 5;
```

The POWER clause specifes the parallelization level for the REBALANCE command. In other words, the POWER clause controls the speed of a rebalance operation. The higher the POWER clause value, the faster ASM will complete the disk rebalancing operation. The default for POWER is 1 (the default value for the ASM_POWER_LIMIT parameter). By specifying the POWER clause in a REBALANCE command, you can override the power value set by the ASM_POWER_LIMIT initialization parameter.

exam watch

You can increase the speed of a rebalancing operation, and thus minimize the I/O impact of modifying ASM storage (by adding or deleting disks), by doing any of the following things: raising the value of the ASM_POWER_LIMIT initialization parameter; using a high value for the POWER clause in a disk rebalance operation; or performing all your disk adding, resizing, and dropping operations at the same time.

CERTIFICATION OBJECTIVE 10.04

Managing ASM Files

ASM doesn't create any datafiles like the datafiles you create when using operating system files. When you use ASM file management, any datafile you create will become an ASM file. You simply specify an ASM disk group, instead of a file or disk, when you want to create a new datafile, redo log file, or control file. For example, you may create a new tablespace called test_tbsp with the following command, if you're using ASM disk groups:

```
SQL> create tablespace test_tbsp  datafile '+test_group1';
```

on the job

You can't use ASM to create alert, trace, binary, and password files.

In the previous command, the DATAFILE clause specifies a file type, indicating that the file is going to be used as a datafile (rather than as a control file or online redo log file, for example). The CREATE TABLESPACE command addresses a disk group (test_group1), not a specific disk in the group. Indeed, there is no reference to any specific datafile either.

ASM does create a datafile, but it is meaningless to compare it with datafiles that you now use in your Oracle databases. The new ASM datafile will be spread over all the disks in the disk group test_group1. Thus, you can't back up a single disk to copy the datafile. Unlike conventional database files, Oracle always creates every ASM file with a specfic redundancy level and striping policy. These redundancy and striping policies are set *permanently* for the files, and you specify the attributes when you create disk groups, as you've seen earlier in this chapter.

exam watch

Any database files that you create by providing a user alias aren't automatically deleted by ASM (when Oracle considers them ready for deletion), since they aren't deemed OMF files. Only files created by ASM will be automatically deleted.

All ASM files are OMF files, and Oracle will automatically delete them when you don't need them any longer. However, if you specify a user alias for an ASM file, that file in't considered an OMF file, so Oracle can't automatically delete that file.

Let's look at some special ASM file management issues in the following sections.

Types of ASM Filenames

In an ASM file system, you never need to know a filename when assigning that file to a tablespace or some other object. You just refer to a disk group—that's it! ASM automatically generates the filenames. When you use ASM for an Oracle database file, the operating system can't see these files, but the RMAN and Oracle's other tools can view them.

To administer the files, you don't need to know their individual names; you just need to know the name of the disk group that contains the file. Of course, if you issue a command like `ALTER DATABASE BACKUP CONTROLFILE TO TRACE`, you'll see the actual name of the ASM files in the output. If ASM uses a *fully qualified name* for a datafile, you can see it in views like `V$DATAFILE` and `V$LOGFILE`.

on the job

You can't use an ASM file system for administrative files like trace files, audit files, alert logs, backup files, export files, tar files, and core files. Oracle stores the ASM filenames in the control files and the RMAN recovery catalog, just as it does with regular operating system-based files or OMF-based files.

You can name an ASM file in several ways, depending on whether you are creating a file or referencing an already existing file. The naming convention may also depend on whether you creating a single files or multiple files at once. The four main ASM filenaming conventions are fully qualified ASM filenames, numeric ASM filenames, alias ASM filenames, and incomplete ASM filenames. Here is an overview of how they work:

- You use fully qualified ASM filenames only for referencing existing ASM files.

- You use numeric ASM filenames only for referencing existing ASM files.
- You use alias ASM filenames for creating new ASM files, as well as referring to existing ASM files. You use alias filenames with templates only for creating new ASM files.
- You use incomplete filenames (with or without an alias) only for file creation operations.

The following sections describe the filename types in more detail.

Fully Qualified ASM Filenames

When ASM creates a file, it always uses a *fully qualified filename*. You use this fully qualified name for *referencing* existing ASM files. Here's the syntax of an ASM file using a fully qualified filename:

```
+group/dbname/file_type/tag.file.incarnation
```

where:

- *`group`* is the *disk group* name.
- *`dbname`* is the name of the *database* to which the Oracle file belongs.
- *`file_type`* shows the ASM *file type*, which maps to an Oracle file type. For example, the ASM file type `controlfile` maps to an Oracle control file. Similary, the ASM `datafile` and `online_log` file types map to Oracle datafiles and online redo logs, respectively.
- *`tag`* is type-specific *information about the file*, such as the tablespace name for a datafile. A control file, for example, is mapped to the ASM tag `CF` (or `BCF`, if it's a backup control file). A datafile's tag is always of the format *ts_name*>_<*file*#>. An online redo log is mapped to the tag *log*_<*thread*#>.
- *`file.incarnation`* is the file/incarnation number pair, used to ensure uniqueness.

on the job

You can't supply a fully qualified ASM filename while creating a new file.

Here's an example of a fully qualified ASM filename:

```
+dgroup1/proddb/controlfile/CF.257.1
```

Realize that no matter what naming convention you use for an ASM file, Oracle will always automatically generate a fully qualified ASM filename when you create a file.

on the job

Oracle also refers to a fully qualified ASM filename as a* system alias. *This is because ASM will create and maintain these aliases, and you can't modify them.

Numeric ASM Filenames

ASM derives *numeric filenames* from fully qualified ASM filenames and uses them to *refer* to existing files. ASM doesn't use numeric filenames in file creation and doesn't use these names while reporting file information to you. However, you can use these abridged names to refer to any file. Here's the form of a numeric filename:

```
+group.file.incarnation
```

Here is an example of a numeric ASM filename (the file number is 251 and the incarnation number is 8675309).

```
+dgroup2.251.8675309
```

Alias ASM Filenames

You can use ASM *alias files* both when *creating* new ASM files and when *referring* to existing files. The only thing common to alias ASM filenames and the previous two types of ASM filenames is the disk group name. In an alias ASM filename, once you first provide the disk group name, you can provide your own *name string* to complete the filename. You can easily tell an alias ASM filename from the previously defined filenaming conventions because there aren't any dotted pairs of numbers at the end of an alias ASM filename. Here's an example of an alias ASM filename:

exam watch

Make sure you understand that alias ASM filenames mean that the files aren't OMF-managed files. Thus, Oracle won't automatically remove these files when it doesn't have any further need for them.

```
+dgroup1/myfiles/control_file1
+dgroup2/mydir/second.dbf
```

Incomplete ASM Filenames

You can use an *incomplete ASM filename* only when *creating* files. Incomplete filenames include just the group name and nothing else. Here's an example of an incomplete ASM filename:

```
+dgroup1
```

You may also use a template while using an incomplete ASM filename, as shown in the following example.

```
+dgroup1(datafile)
```

In this example, the template, DATAFILE, is within the parentheses.

on the job

A template determines the attributes to be applied to a file when you create a new file.

exam watch

When you use an incomplete ASM filename, the use of a template is optional. However, if you omit the template, ASM will use the default system template for that file type to decide the redundancy and striping characteristics for that file. For example, when creating a datafile for a tablespace, ASM will use the default template for a datafile file type if you don't provide a specific template in the filename.

Alias Filename Management

When you review the ASM files later in this chapter, you'll see how Oracle automatically generates fully qualified names for all the ASM files you create in your database. Alias ASM filenames provide the capability to employ user-friendly filenames to substitute for the cryptic system-generated filenames.

Creating Disk Group Directories for Alias Filenames

Oracle maintains a hierarchical directory structure for all the fully qualified filenames in each disk group (along with any file aliases you may create, as shown later in this chapter). If you intend to use alias ASM filenames, you must first create a *directory structure* to support your alias filenaming conventions. For example, the following statement creates a hierarchical directory for disk group dgroup1.

```
alter diskgroup dgroup1 add directory '+dgroup1/mydir';
```

Once you create the directory dgroup1/mydir as shown in the example, you can use it to create alias ASM filenames, such as `+dgroup1/mydir/control_file1` for a control file.

Using Templates with Aliases

You can also use a *template* to create an alias ASM filename. You can use template-based ASM filenames only for creating new files. Here's the syntax of a template-based alias ASM filename:

```
dgroup(template_name)/alias
```

Here's an example of an alias ASM filename with a template (where the template name is SPFILE):

```
+dgroup1(spfile)/config1
```

Adding Aliases

If you don't use an alias for a file at the file creation time, you can always modify a filename later to add an alias for the file. You can add a filename alias or rename an existing alias name, using the `ADD ALIAS` or `RENAME ALIAS` clause of the `ALTER DISKGROUP` statement. Here's an example that shows how you can replace a fully qualified ASM filename with your own alias ASM filename:

```
alter diskgroup dgroup1 add alias '+dgroup1/mydir/second.dbf'
for '+dgroupA/sample/datafile/mytable.342.3';
```

You can also delete an alias by using the `DROP ALIAS` clause of the `ALTER DISKGROUP` statement.

Dropping Files and Aliases from a Disk Group

Although ASM files are usually OMF files, there may be times when you use your own aliases for some ASM files. As mentioned earlier, if you use your own aliases for ASM files, those files won't be automatically deleted by Oracle. In situations like this, you'll then need to use the `DROP FILE` command to drop those files and aliases.

Here is an example of dropping an alias name:

```
alter diskgroup dgroup1 drop file '+dgroup1/payroll/compensation.dbf';
```

The following is an example of using a system-generated filename to drop a datafile:

```
alter diskgroup dgroup1
drop file '+dgroupA/sample/datafile/mytable.342.372642';
```

on the job

If you just drop a file alias, Oracle won't drop the file. If you drop a file using the `DROP FILE` command, however, Oracle drops both the file and its alias.

ASM Filename Usage

You have seen how you can use several forms of ASM filenames. You've also learned how you can use different filenames for the same file. *When* do you use a particular form of a filename? The correct form of the ASM filename you should use depends on the context in which you are using the filename. Here's a brief summary of when you should use a particular type of ASM filename:

- For *referring to an existing file*, you can use a fully qualified filename, a numeric filename, or an alias filename (but not an alias name with a template or an incomplete filename with or without a template).
- For *creating a single file*, you can use any kind of filename, with the exception of a fully qualified filename.
- For *creating multiple files*, you use only an incomplete filename or an incomplete filename with a template.

Since the very purpose in using ASM is to eliminate the need to specify filenames when managing database files, you must avoid using the ASM filenames as much as possible, even though most SQL commands will let you use them just like normal filenames. For example, when you re-create a control file with the RESETLOGS option, you may use the ASM filenames, as shown in the following example:

```
create controlfile reuse database  "TEST" resetlogs archivelog
    maxlogfiles 16
    maxlogmembers 2
    maxdatafiles 30
    maxinstances 1
    maxloghistory 226
logfile
  group 1 ('+DGROUP1','+DGROUP2') size 100M,
  group 2 ('+DGROUP1','+DGROUP2') size 100M
datafile
  '+DGROUP1/db/datafile/system.260.3' size 500M,
  '+DGROUP1/db/datafile/sysaux.259.3' size 500M
```

In the previous example, note that the datafiles already exist, so you can use the fully qualified ASM filenames for the datafiles. As you know, fully qualified ASM filenames can't be used to *create* an ASM file—they can be used only to *refer* to ASM files. The RESETLOGS option will re-create (reinitialize) the redo log files, so you can use the incomplete ASM filename format for the redo log files. You can use incomplete filenames only to create new ASM files, as shown earlier in this chapter.

ASM File Templates

You can't create all ASM files with the same file attributes. *File attributes* in this context refer to attributes like the redundancy level (external, normal, or high) and the striping format (fine or coarse). For example, ideally, the file attributes of an online redo log file must be different from those of datafiles. Oracle makes it easy for you to specify file attributes by letting you use templates to specify the attributes when you are creating files. These templates are applied to the individual files, but they are associated with the disk group in which the files are created.

Whenever you create a disk group, Oracle establishes a set of initial system default templates for that disk group. Each of the system templates will contain a set of specific file attributes. When you create files under this disk group, you can choose from among these Oracle-provided system templates. Table 10-1 shows some system templates and the file attributes they imply.

An example will make the use of ASM file templates clear. Let's say you want to create a new tablespace called test_tbsp in an ASM file system. You know that the tablespace will use datafiles (not online logs or control files). You thus use the DATAFILE template for your tablespace:

```
SQL> create tablespace  test_tbsp  datafile '+test_group1';
```

Your tablespace datafile will then inherit the default attributes of a `DATAFILE` template (such as the coarse striping level).

You may alter the attributes of a default system template or create you own unique templates, if you wish. However, you may not delete the default system templates.

The following example shows how you can create your template, called PRODUCTION, using the `ALTER DISKGROUP` command.

```
SQL> alter diskgroup test_group1 add template production attributes (mirror fine)
```

TABLE 10-1 ASM Default File Group Templates

Template Name	File Type	External Redundancy	Normal Redundancy	High Redundancy	Striped
CONTROL	Control files	Unprotected	2-way mirror	3-way mirror	Fine
DATAFILE	Datafiles and copies	Unprotected	2-way mirror	3-way mirror	Coarse
ONLINELOG	Online logs	Unprotected	2-way mirror	3-way mirror	Fine
ARCHIVELOG	Archive logs	Unprotected	2-way mirror	3-way mirror	Coarse

Once you create the production template, you can create an ASM datafile using that template. The datafile you create will inherit the attributes you specified for your new PRODUCTION template (two-way mirroring and fine striping). Here's how you would create your new datafile using the PRODUCTION template (the disk group test_group1 creates files based on the PRODUCTION template):

```
SQL> create tablespace test_tbsp2 datafile '+test_group1';
```

If you want to drop a template, you may do so with the following command:

```
SQL> alter diskgroup  test_group1 drop template production;
```

exam watch

A template's main function is to simplify ASM file creation. Templates are associated with a disk group, but they are applied to individual files. You can't change a files's attributes once you create it using a certain template. If you wish to change an ASM file's attributes, you must use the RMAN to copy the file into a new file with the attributes you want.

CERTIFICATION OBJECTIVE 10.05

Migrating a Database to ASM

Migrating your database to an ASM disk group-based storage system database is one of the Oracle Database 10*g* upgrade certification objectives. Before we get to the actual database migration procedures, let's look at the necessary ASM-related initialization parameter changes.

Setting Instance Parameters

You need to focus on the following initialization parameters when you create an ASM-based Oracle database:

- **INSTANCE_TYPE** When you create the ASM instance, the INSTANCE_TYPE parameter is set to ASM. However, for normal databases, you can either set this to RDBMS or leave it out, since the default value for this parameter is RDBMS.

- **`DB_BLOCK_SIZE`** The value of the `DB_BLOCK_SIZE` parameter must be set to one of the standard block sizes: 2KB, 4KB, 8KB, 16KB, or 32KB.
- **`LOG_ARCHIVE_FORMAT`** If you set the `LOG_ARCHIVE_FORMAT` to an incomplete ASM filename (such as +dgroupA), Oracle will ignore it. If you set it to an ASM directory, Oracle will use the directory and create non-OMF files in that directory.

You must use *incomplete* ASM filenames (or incomplete ASM filenames with a template) as the destination for the following initialization parameters:

- DB_CREATE_FILE_DEST_*n*
- DB_CREATE_FILE_DEST
- DB_RECOVERY_FILE_DEST
- CONTROL_FILES
- LOG_ARCHIVE_DEST_*n*
- LOG_ARCHIVE_DEST
- STANDBY_ARCHIVE_DEST

Creating an ASM-Based Database

By specifying the following parameters, you can create an ASM disk group-based Oracle database:

```
DB_CREATE_FILE_DEST = '+dgroup1'
DB_RECOVERY_FILE_DEST = '+dgroup2'
DB_RECOVERY_FILE_DEST_SIZE = 100G
```

The database creation statement is very simple, as shown here:

```
SQL> CREATE DATABASE test;
```

Oracle will create a SYSTEM tablespace and a SYSAUX tablespace in the disk group dgroup1. An undo tablespace will also be created in dgroup1, if you've configured automatic undo space management. A multiplexed redo file log group and a control file will be created in both group1 and group2.

Adding datafiles is very easy in an ASM-based Oracle database, since you don't need to specify the datafiles. In the test database, you can use the following commands to create a tablespace and add a new redo log file, respectively:

```
create tablespace test_tbsp;
alter database add logfile;
```

Migrating Your Database to ASM

The RMAN can help you migrate your current database to an ASM storage system. If you want to migrate your current database to ASM disk groups, you may do so by using the following steps (this example uses an SPFILE):

1. Shut down the database in a consistent mode (not `SHUTDOWN ABORT`).
2. Make the necessary changes to your SPFILE, so your database can use OMF files. Set the OMF destinations for the initialization parameter you reviewed in the previous section to their appropriate ASM destinations (ASM disk groups).
3. Delete the control file parameter from your SPFILE, since Oracle will automatically create new control files by restoring them from the non-ASM instance control files.
4. Start up the database using the command `STARTUP NOMOUNT`, and then execute the following commands (as an RMAN script):

   restore controlfile from '/u01/test/c1.ctl';

   alter database mount;

   backup as copy database format '+dgroup1';

   switch database to copy;

```
SQL "alter database rename '/u01/test/log1' to '+dgroup1' "; -- for
    each redo log member

SQL> "alter database open resetlogs";
SQL "alter tablespace  temp add tempfile" -- for each temporary tablespace
SQL "alter database tempfile '/u01/temp1' drop";
```

5. The RMAN script will back up the database and switch current datafiles to the backups. The script also renames all current online redo log files and re-creates all temporary files for the temporary tablespace.
6. Delete all the old Oracle database files.

exam watch

When you migrate to an ASM database, Oracle will automatically create the control files for you. Oracle will rename the existing redo log files and re-create all the temporary files for the temporary tablespace.

Using the Data Dictionary to Manage ASM

There are several new data dictionary views in Oracle 10*g* to help you manage Automatic Storage Mangement. Let's briefly review these new views (the first three views have rows for both the ASM instance as well as your database instances).

V_ASM_DISKGROUP

- In an ASM instance, this view provides information about a disk group. In a database instance, this view contains one row for every ASM disk group mounted by the ASM instance.

V_ASM_CLIENT

- In an ASM instance, this view identifies all the client databases using various disk groups. In a Database instance, the view contains one row for the ASM instance if the database has any open ASM files.

V$ASM_DISK

- In an ASM instance, this view contains one row for every disk discovered by the ASM instance. In a database instance, the view will only contain rows for disks in use by that database instance.

V$ASM_FILE

- The V$ASM_FILE view contains one row for every ASM file in every disk group mounted by the ASM instance.

V$ASM_TEMPLATE

- This view contains one row for every template present in every disk group mounted by the ASM instance.

V$ASM_OPERATION

- The V$ASM_OPERATION view provides information about any active long-running operations in the ASM instance. Several ASM commands like `REBALANCE` and `RESIZE`, take considerable time to complete. The command prompt may return right away when you use these commands, however. You can use the `V$ASM_OPERATION` view to monitor these types of long-running operations.

INSIDE THE EXAM

The exam concentrates evenly on all aspects of using ASM. From a conceptual point of view, you need to understand the relationship between ASM files and disk groups on one hand, and Oracle database files on the other. You must be able to answer questions on what types of Oracle files you can store in an ASM storage system. Pay particular attention to the ASM instance parameter file and know the importance of each of the initialization parameters. The test focuses on the new background processes in ASM-based systems, as well as the relationship between disk groups and failure groups.

The test will probe into your knowledge of the various startup and shutdown commands in an ASM-based system. Know the relationship between shutting down an ASM instance and the running of the connected Oracle instances. For example, what happens to the connected Oracle instances if you abort an ASM instance? Rebalancing of disks in a disk group is a significant ASM detail, and expect a question regarding rebalancing operations or parameters.

You must understand how to refer to ASM files in different situations (for example, while creating a new control file). File aliases and file templates are also bound to show up in one or more questions. Know how you use different filenaming syntax for various purposes (creating and referring to files). What are file templates, what do you define them on, and why do you use them?

The test may also contain questions regarding migrating your current database to an ASM-based system. You must clearly understand the steps of the migration process. Important points to remember are how Oracle handles the control file redo log files and datafiles for the newly converted database. In other words, how does the database create the new files?

CHAPTER SUMMARY

This chapter reviewed the new Automatic Storage Management (ASM) feature of Oracle Database 10*g*. ASM builds on the OMF files concept and now makes it possible to manage large databases without needing to worry about naming database files. The chapter showed you how managing disk groups is a far more efficient way of managing storage than handling storage at the operating system file level.

You also learned how ASM rebalances files automatically when you add or remove disks from disk groups. You learned how to configure and manage an ASM instance.

You then learned how to create and manage disk groups. ASM files use special naming conventions, and you learned about the various types of ASM filenames in this chapter. Toward the end of this chapter, you saw how to create ASM-based databases. You also learned how to migrate your current databases to an ASM-based storage system.

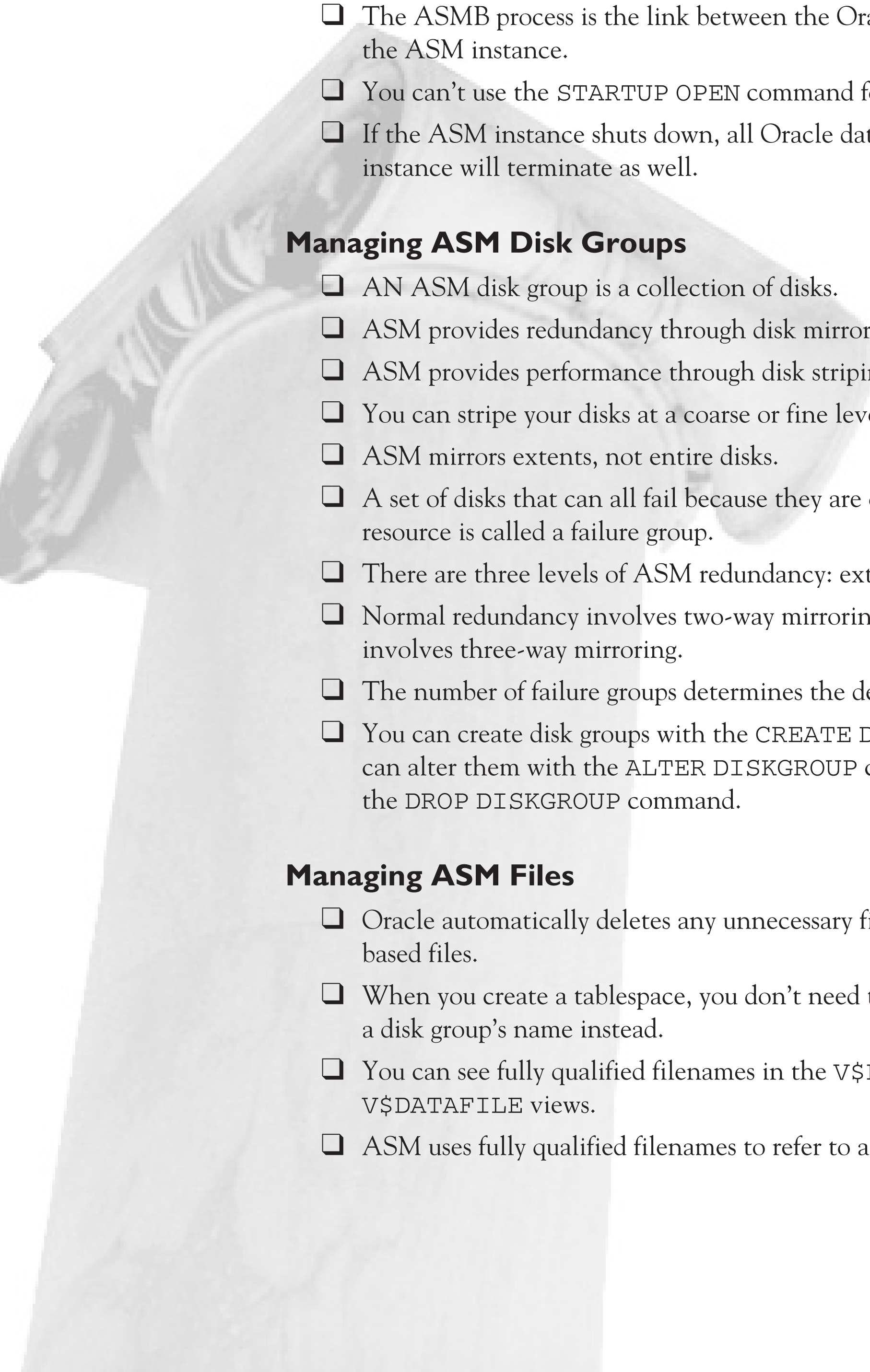
✓ TWO-MINUTE DRILL

Introduction to Automatic Storage Management

- ❑ ASM automates and optimizes database storage for optimal performance.
- ❑ ASM is built on top of OMF files.
- ❑ You need to use the RMAN to copy ASM files.
- ❑ You can mingle ASM, OMF, and operating system files together.
- ❑ ASM simplifies storage management because you deal with only a few disk groups, not a large number of disks and datafiles.
- ❑ ASM automatically balances I/O load.
- ❑ ASM helps you maintain redundant copies of data to provide fault tolerance.
- ❑ You access ASM files by accessing ASM disk groups.

Managing the ASM Instance

- ❑ You must have the SYSOPER privilege to create an ASM instance.
- ❑ The ASM instance doesn't mount files. It mounts disk groups.
- ❑ The ASM instance's main job is to maintain ASM file metadata.
- ❑ The ASM instance is in charge of communicating file metadata to the database instances.
- ❑ In order to use ASM, the ASM instance must be running.
- ❑ Your database never connects directly to ASM disk groups.
- ❑ You must issue all disk group management commands from an ASM instance.
- ❑ The only initialization parameter that you must provide for an ASM instance is the `INSTANCE_TYPE` parameter.
- ❑ The `INSTANCE_TYPE` parameter must be set to `ASM` for starting an ASM instance.
- ❑ The `ASM_POWER_LIMIT` parameter determines the speed of a rebalance operation.
- ❑ The `ASM_DISKSTRING` parameter limits the disk-discovery process.
- ❑ The ASM instance uses the `LARGE_POOL` memory.
- ❑ The ASM instance has two new background processes: RBAL and ARB*n*.



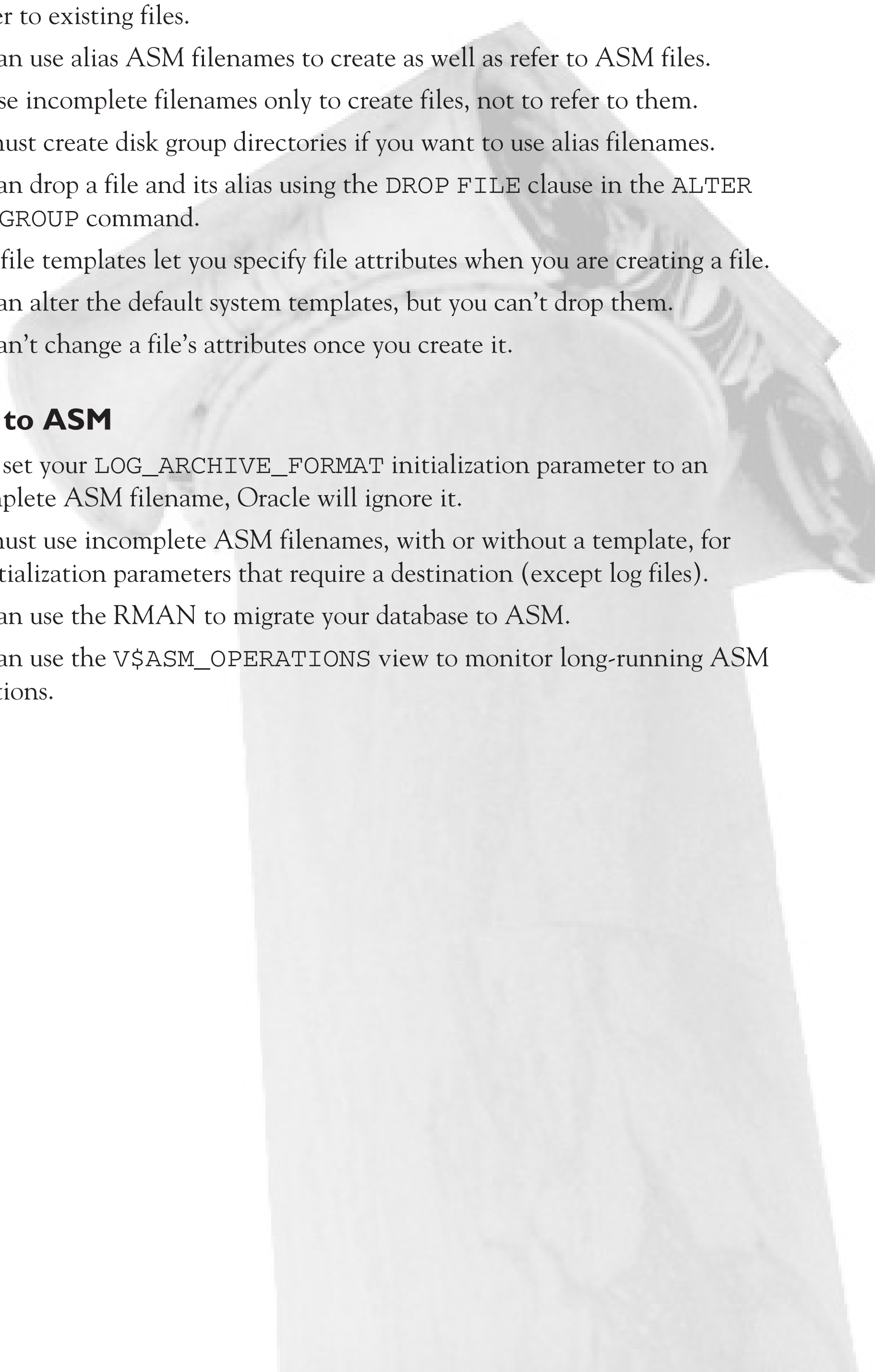
- ❑ The Oracle instance using ASM will have two new processes: RBAL and ASMB.
- ❑ The ASMB process is the link between the Oracle database instance and the ASM instance.
- ❑ You can't use the `STARTUP OPEN` command for an ASM instance.
- ❑ If the ASM instance shuts down, all Oracle databases connected to the instance will terminate as well.

Managing ASM Disk Groups

- ❑ AN ASM disk group is a collection of disks.
- ❑ ASM provides redundancy through disk mirroring.
- ❑ ASM provides performance through disk striping.
- ❑ You can stripe your disks at a coarse or fine level.
- ❑ ASM mirrors extents, not entire disks.
- ❑ A set of disks that can all fail because they are connected to a common resource is called a failure group.
- ❑ There are three levels of ASM redundancy: external, normal, and high.
- ❑ Normal redundancy involves two-way mirroring, and high redundancy involves three-way mirroring.
- ❑ The number of failure groups determines the degree of redundancy.
- ❑ You can create disk groups with the `CREATE DISKGROUP` command. You can alter them with the `ALTER DISKGROUP` command and drop them with the `DROP DISKGROUP` command.

Managing ASM Files

- ❑ Oracle automatically deletes any unnecessary files only if they are OMF-based files.
- ❑ When you create a tablespace, you don't need to specify a datafile. Just use a disk group's name instead.
- ❑ You can see fully qualified filenames in the `V$LOGFILE` and `V$DATAFILE` views.
- ❑ ASM uses fully qualified filenames to refer to a file.



- ❑ A fully qualified filename is also called a system alias.
- ❑ ASM derives numeric filenames from fully qualified filenames and uses them to refer to existing files.
- ❑ You can use alias ASM filenames to create as well as refer to ASM files.
- ❑ You use incomplete filenames only to create files, not to refer to them.
- ❑ You must create disk group directories if you want to use alias filenames.
- ❑ You can drop a file and its alias using the `DROP FILE` clause in the `ALTER DISKGROUP` command.
- ❑ ASM file templates let you specify file attributes when you are creating a file.
- ❑ You can alter the default system templates, but you can't drop them.
- ❑ You can't change a file's attributes once you create it.

Migrating to ASM

- ❑ If you set your `LOG_ARCHIVE_FORMAT` initialization parameter to an incomplete ASM filename, Oracle will ignore it.
- ❑ You must use incomplete ASM filenames, with or without a template, for all initialization parameters that require a destination (except log files).
- ❑ You can use the RMAN to migrate your database to ASM.
- ❑ You can use the `V$ASM_OPERATIONS` view to monitor long-running ASM operations.

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, because there might be more than one correct answer. Choose all correct answers for each question.

Introduction to Automatic Storage Management

1. Which of the following statements about ASM instances is true?
   - A. ASM instances need to be open before Oracle can access ASM database files.
   - B. ASM instances need to be mounted before Oracle can access ASM database files.
   - C. ASM instances must be started, but in an unmounted state, before Oracle can access ASM database files.
   - D. ASM instances need to be shut down before Oracle can access ASM database files.
2. You must use which type of commands to copy ASM files?
   - A. RMAN commands
   - B. OMF commands
   - C. Operating system commands
   - D. Logical Volume Manager commands
3. Which types of files can you use in a single database?
   - A. Either ASM or OMF files
   - B. OMF, ASM, and operating system files
   - C. OMF and ASM files
   - D. ASM and Logical Volume Manager-based operating system files
4. ASM performs load balancing by distributing which items?
   - A. File extents across all disks in a disk group
   - B. ASM disks across ASM disk groups
   - C. File extents across all disks in all disk groups
   - D. Heavily used tables uniformly across all disks in a disk group

Managing the ASM Instance

5. What are the functions of an ASM instance?
   - A. Managing disk groups and communicating file metadata to the Oracle database using ASM files

   - B. Managing database files and communicating file metadata to the ASM instance using ASM files
   - C. Managing disk groups and communicating file metadata to the RMAN
   - D. Protecting disk groups

6. What is the only initialization parameter that you absolutely must set to create an ASM instance?

   - A. `DB_UNIQUE_NAME`
   - B. INSTANCE_TYPE
   - C. ASM_POWER_LIMIT
   - D. ASM_DISKSTRING

7. What happens if you don't set the `ASM_DISKSTRING` parameter for an ASM instance?

   - A. Your ASM instance won't start.
   - B. Your Oracle database won't be able to communicate with the ASM instance.
   - C. The disk-discovery process may take a longer time when you add a disk to a disk group.
   - D. ASM won't perform disk discovery when you add a new disk to a disk group.

8. What does the ARB*n* background process do in an ASM instance?

   - A. Coordinates disk activity
   - B. Performs disk rebalance work
   - C. Coordinates disk rebalance work
   - D. Manages the RBAL process

9. When you start an ASM instance in the `MOUNT` mode, what does the instance mount?

   - A. Oracle database files
   - B. Individual ASM disks that belong to a particular database
   - C. Disk groups specified by the `ASM_DISKGROUPS` parameter
   - D. The instance doesn't have any database files, so it doesn't mount anything

Managing ASM Disk Groups

10. What do ASM disk groups provide?

   - A. Redundancy through striping
   - B. Performance through mirroring
   - C. Redundancy through mirroring
   - D. Performance through striping

10. What does ASM stripe?

- **A.** Database files across all the disk groups
- **B.** ASM disks across all the disk groups
- **C.** ASM disk groups across all the database files
- **D.** Database files across all the disks in a disk group

12. What does ASM mirror?

- **A.** Disks
- **B.** Files
- **C.** Extents
- **D.** Disk groups

13. For two-way mirroring, how many failure groups do you need at a minimum?

- **A.** None (two-way mirroring doesn't need any failure groups)
- **B.** One failure group
- **C.** Two failure groups
- **D.** Three failure groups

Managing ASM Files

14. Which of the following types of files will the database automatically delete?

- **A.** All unnecessary files
- **B.** All unnecessary ASM files without aliases
- **C.** All unnecessary ASM files with aliases
- **D.** All unnecessary ASM files with and without aliases

15. When can you use a fully qualified name?

- **A.** Only to create a new ASM file
- **B.** Only to refer to an ASM file
- **C.** To create and to refer to an ASM file
- **D.** For only ASM system files

16. What are incomplete filenames used for?

- **A.** Referring to multiple files
- **B.** Creating single files
- **C.** Creating multiple files
- **D.** Creating ASM system files

17. Which of the following is true about changing an ASM file's attributes?

A. You must use the `ALTER DISKGROUP` command to alter the template.

B. You must use the `ALTER DISKGROUP` command to alter the file attributes directly.

C. You must use a new file alias for the file before altering its attributes.

D. You can't change a file's attributes after you create it.

Migrating to ASM

18. If you set the `LOG_ARCHIVE_FORMAT` initialization parameter to an ASM directory, what will Oracle do?

A. Oracle will use the directory and create non-OMF files in that directory.

B. Oracle will use the directory and create OMF files in that directory.

C. Oracle will not use that directory, unless the online redo log files are also in an ASM directory.

D. Oracle will use the directory only if you use an incomplete ASM filename.

19. When you migrate your current database to ASM-based storage, what should you do with your redo log files?

A. Rename them to ASM disk groups

B. Reinitialize the current online redo logs

C. Oracle will automatically copy them when you migrate to an ASM-based database

D. Copy them from their present locations to the ASM disk directly

20. What is the most important thing that the `V$ASM_OPERATION` view helps you do?

A. Monitor the success or failure of disk-balancing operations

B. Monitor long-running operations

C. Monitor the migration to an ASM instance

D. Monitor all disk groups to see if disk rebalancing is necessary

LAB QUESTIONS

1. Show how you would add a disk to an existing disk group.

2. Show how you would remove an existing disk group.

3. Show how you would undo the pending disk removal operation.

4. Show all your ASM disks using a data dictionary view.

5. Show all your ASM files using a data dictionary view.

6. Create a tablespace using an ASM disk group.

SELF TEST ANSWERS

Introduction to Automatic Storage Management

1. ☑ **B.** The ASM instance must be mounted before Oracle can access the ASM database files. When you start the ASM instance in the `MOUNT` mode, the instance mounts all the ASM disk groups. Without accessing the disk groups, you can't access the ASM datafiles that are on the ASM disks.
   ☒ **A** is wrong because an ASM instance cannot ever be opened like an Oracle database instance. **C** is wrong because if the ASM instance isn't mounted, you can't access the ASM disk groups. **D** is wrong since the ASM instance must be running for Oracle to access its ASM-based database files.

2. ☑ **A.** You can copy ASM files only with RMAN commands.
   ☒ **B** is wrong because there are no OMF commands to copy files. **C** is wrong since the operating system doesn't "see" ASM files, so it can't manage them. **D** is wrong because you don't need a Logical Volume Manager to use with your ASM files.

3. ☑ **B.** You can use all three types of files—OMF, ASM, and operating system files—in the same database.
   ☒ **A, C,** and **D** specify the wrong combinations of files.

4. ☑ **A.** ASM performs I/O load balancing by distributing file extents uniformly across all disks in a disk group.
   ☒ **B** is wrong since ASM doesn't perform load balancing through distributing disks across disk groups. **C** is wrong because the file extents are spread only across disks in the same disk group. **D** is wrong because ASM doesn't deal with database tables in any way.

Managing the ASM Instance

5. ☑ **A** and **D.** The key job of the ASM instance is to manage disk groups. It also communicates information about the disk group to the Oracle database using ASM files. Another key task managed by the ASM instance is protecting disk groups .
   ☒ **B** is wrong because ASM doesn't manage database files directly—that job falls to the database. **C** is wrong because ASM doesn't communicate file metadata to the RMAN.

6. ☑ **B.** The only ASM initialization parameter that you must set is the `INSTANCE_TYPE` parameter. You must set it to a value of `ASM` to indicate to the Oracle executable that you are starting an ASM instance, not a regular Oracle database instance.

☒ **A, C,** and **D** are wrong because none of these parameters are necessary to start an ASM instance. The instance can use the default values for all three parameters in order to start.

7. ☑ **C.** The `ASM_DISKSTRING` parameter specifies the location of the ASM disk groups. If you don't specify a value for this parameter, the disk-discovery process undertaken by ASM instance will take a longer time, since it will look in all directories to which the ASM instance has read/write access.
☒ **A** is wrong because the ASM instance will start, even when you don't specify a value for the `ASM_DISKSTRING` parameter. **B** is wrong because the ASM instance will communicate normally with the Oracle database. **D** is wrong because ASM will perform the disk-discovery process, but it may take a longer time to do so if you omit the `ASM_DISKSTRING` parameter.

8. ☑ **B.** The ARB*n* background process performs the actual disk rebalancing work in an ASM instance.
☒ **A** and **C** are wrong since it is the RBAL background process that coordinates the disk activity, not the ARB*n* process. **D** is wrong because the RBAL process manages the ARB*n* process, not the other way around.

9. ☑ **C.** When you mount an ASM instance, the instance mounts the disk groups you specify by using the `ASM_DISKGROUPS` initialization parameter.
☒ **A** is wrong because ASM instances don't mount Oracle database files. **B** is wrong since an ASM instance mounts disk *groups*, not individual *disks* that are part of the disk group. **D** is wrong since the ASM instance does mount disk groups.

Managing ASM Disk Groups

10. ☑ **C** and **D.** Mirroring provides redundancy, and striping enhances performance.
☒ **A** and **B** are wrong since they mix up the functions of mirroring and striping.

11. ☑ **D.** ASM stripes database files across all the disks in a disk group.
☒ **A** is wrong because ASM doesn't stripe database files across multiple disk groups. **B** and **C** are wrong since ASM doesn't stripe disks or disk groups—it stripes database files.

12. ☑ **C.** ASM mirroring is different from operating system mirroring in that it mirrors individual extents, not disks.
☒ **A, B,** and **D** refer to the wrong unit for ASM mirroring.

13. ☑ **C.** You need two failure groups at a minimum for two-way mirroring.
☒ **A** is wrong because you do need failure groups for any mirroring. **B** is wrong because a single failure group doesn't ensure protection of any kind. **D** is wrong because three failure groups provide *three-way* mirroring.

Managing ASM Files

14. ☑ **B.** The database will automatically delete all ASM files without aliases, because they are considered OMF files.
☒ **A** is wrong because the database will automatically delete only the OMF files, not all the unnecessary files. **B** and **D** are wrong since Oracle considers files with aliases non-OMF files, and therefore, it won't automatically delete them.

15. ☑ **B.** You can use a fully qualified name only to *refer* to an ASM file.
☒ **A** and **C** are wrong since you can't use fully qualified names to create files. **D** is wrong since you can use fully qualified names for nonsystem files.

16. ☑ **C.** Incomplete ASM filenames are used only for creating multiple files.
☒ **A** is wrong because incomplete filenames can't be used to merely refer to files. **B** is wrong because incomplete filenames are not limited to creating single files. **D** is wrong since incomplete filenames aren't limited to system files.

17. ☑ **D.** You can't change a file's attributes once you create it.
☒ **A, B,** and **C** are wrong since you can't change a file's attributes after creating it.

Migrating to ASM

18. ☑ **A** and **D. A** is correct because Oracle will create non-OMF files in that directory. **D** is correct because Oracle will use the directory only if it has an incomplete ASM filename.
☒ **B** is wrong since Oracle will create non-OMF files in that directory. **C** is wrong since Oracle will use that directory.

19. ☑ **A.** When you migrate your database to ASM files, you must individually rename each of your online redo log files to ASM disk groups.
☒ **B** is wrong since you don't reinitialize your online redo log files, as this would clear them. **C** is wrong since Oracle will not automatically copy your online redo log files. **D** is wrong since you copy the files to disk groups, not directly to the underlying disks.

20. ☑ **B.** The most important function of the `V$ASM_OPERATION` view is to help you monitor long-running ASM operations.
☒ **A, C,** and **D** are wrong since the `V$ASM_OPERATION` doesn't help you monitor any of these things.

Lab Answers

1. To add a disk to an existing disk group, use the following:

```
SQL> alter diskgroup dgroup1
     add failgroup controller1
     '/dev/rdsk/c0t3d0s2' NAME a5;
```

2. Remove an existing disk group as follows:

```
SQL> drop diskgroup dgroup1 including contents;
```

3. You can undo a pending disk removal operation with this command:

```
SQL> alter diskgroup dgora1 undrop disks;
```

4. To show all your ASM disks using a data dictionary view, use the following:

```
SQL> select name, failgroup, bytes_read, bytes_written
     from v$asm_disk;
```

5. To show all your ASM files using a data dictionary view, use the following:

```
SQL> select  group_number, file_number, bytes, type, striped
     from v$asm_file;
```

6. Create a tablespace using an ASM disk group as follows:

```
SQL> create tablespace  test_tbsp
     datafile '+dgroup1' size 500M
```