CHAPTER 1

Overview of Migration, Transformation, and Upgrade Methodologies
We have written this book to fill a need that we’ve noticed has gone quite unanswered: a practical roadmap for Oracle DBAs and their colleagues on the front lines of database migration, transformation, and upgrades to effectively plan, carefully implement, and successfully complete an upgrade, migration, or transformation project. We’ve classified three different types of projects within this book:

- **Data migration**  This type of project involves moving data from one database to another. It’s not uncommon for a data migration project to share some of the elements of a database upgrade project as well.

- **Database upgrade** We identify two key methodologies for upgrading an Oracle database: the *in-place upgrade*, in which the Oracle binary files are upgraded and scripts are executed to upgrade objects in the database itself; and the *out-of-place upgrade*, in which all the data from the previous version of the database is moved to another database that’s already at a more recent version.

- **Migration with transformation** This type of project typically requires modifications to the database application objects as well as the database itself—for example, implementing a data partitioning scheme for a previously unpartitioned table, or migrating large objects (LOBs) from BasicFile to SecureFile format, or even encrypting previously unsecured data.

Based on our experience, however, not all projects fall into a single category, and there are often overlapping requirements during a data migration and database upgrade project. To add to the confusion, there are multiple migration methods that may or may not be appropriate for each project’s requirements. One of our major goals in this book is to demystify which methods will work for different migration, upgrade, and transformation scenarios, as well as to help you identify which methods are most appropriate for your project’s requirements. This book will also explain that sometimes there is only one option that’s appropriate to accomplish a particular project’s requirements.

**Database Upgrade Methodologies**

In essence, upgrading an Oracle database simply brings it to a newer database release from an older database release. An upgrade can involve upgrading the Oracle software and database (an *in-place upgrade*) or moving to a newer system (a *migration with upgrade*). We typically see more migration-with-upgrade projects than any other project types because many organizations want to take advantage
of improved hardware and storage. Moving a database to a new storage system, operating system, or computing platform is typically an excellent opportunity to upgrade to a newer Oracle database release as well.

It's not often that an upgrade goes bad, but it does happen, so we have found that it's crucial to consider a fallback solution just in case. An advantage of migration with upgrade is that it leaves the original production system intact, thus providing a readily available fallback solution. This should be part of your project planning process, which we will cover extensively in Chapters 2 and 3; we will demonstrate the recommended best practices for using Oracle's Database Upgrade Assistant in Chapter 11.

The following represents an overview of the methods that we cover in this book for performing database upgrades or database upgrades with migrations. Each method is covered in depth in a chapter in the book.

Data Migration Methodologies
In its simplest form, a data migration involves moving data from one database to another. Depending on the method selected, a database migration can optionally include an upgrade as well as data transformation. There are multiple methods of data migration, but which method you should choose for your migration will vary based on your database and your requirements. These methods are described in the following sections.

Data Guard
Migrating with Data Guard is covered in Chapter 4. This is one of several specialty migrations that are available only under a narrow set of circumstances. However, when the requirements fit your needs, migrating with Data Guard can be a very effective and powerful method of performing a data migration. Data Guard typically requires that the platform, database version, and database are identical on both source and target systems.

In its simplest form, this migration begins with the creation of the new system as a Data Guard physical standby database. When it's time to cut over to the new system, you perform a Data Guard switchover operation. The new system becomes the primary database, and the original system is transformed into a ready-made physical standby database.

RMAN
We demonstrate how to use Recovery Manager (RMAN) to perform a database migration in Chapter 5. In an appropriate scenario, RMAN makes it simple to back up a database on one platform and restore it on another platform to complete the migration.
Transporting Tablespaces
One of the most flexible methods for data migration—transporting sets of tablespaces between databases—is covered in Chapter 6. This method allows you to move a tablespace from a source system to a destination system even when the systems are housed on completely different platforms or operating systems.

Cross-Platform Tablespace Migration
The Cross-Platform Transportable Tablespace (XTTS) migration utility—a specialized data migration method that requires extremely limited application downtime—is discussed in detail in Chapter 9.

Export/Import
Export/Import (which also includes Data Pump Export/Import) is covered in Chapter 7. In its simplest form, Export/Import takes data from one database and copies it into another database. Because of the way the export and import process works, it’s also possible to perform data transformation during this process, so data can be imported into a table with a different character set, a different partitioning configuration, and so forth. Because of these properties, Export/Import is the most flexible data movement tool. Export/Import is one of the two migration-with-transformation methods that we cover in this book as well as the basis for the second one: Oracle GoldenGate.

GoldenGate
Performing a migration with Oracle GoldenGate is covered in Chapters 8 and 12. Chapter 8 shows how to perform a minimal downtime migration upgrade and transformation using Oracle GoldenGate, while Chapter 12 shows how to migrate to Oracle from a non-Oracle database—in this case, Microsoft SQL Server.

Oracle GoldenGate is a heterogeneous replication product. It enables you to set up a source system and a target system and keep them synchronized even if they are different Oracle versions, different platforms, or even completely different databases. Migration using GoldenGate is a multiphase operation. First, GoldenGate is set up to capture changes. Next, an initial migration is done. Then GoldenGate applies those changes and keeps the source and target systems in sync. Finally, the cutover is performed. With this method, the cutover is very fast. As we have observed, “It doesn’t matter how long it takes to do the migration ... just how long it takes to do the cutover.”

Automatic Storage Management (ASM)
ASM migrations are covered in Chapter 10. This is another specialty migration method and it is only valid for migrating to new storage. It is a very simple and straightforward process and can be done with no downtime, but again, it only migrates the database to new storage. This method involves bringing in the new
storage system and adding it to your system’s existing ASM disk groups. After the new storage system has been added to ASM, the original disks can be dropped from the disk groups. This method works extremely well—we have performed it many times at many client sites—but it is important to note that it does not allow for any data transformation.

Migration with Transformation Methodologies
Migration with transformation usually involves improving an existing database schema. Because Oracle makes many improvements with each database release, it’s often challenging to take advantage of these improvements with an existing database; sometimes the only opportunity to take advantage of them might be while you are migrating your database.

Some new features available in later Oracle database releases that you might want to consider include

- **Partitioning**  Partitioning has improved tremendously over the last few database releases, especially in Oracle 11.1.0.7, which added four new partitioning methods.

- **LOB storage**  SecureFile LOB storage is a great improvement over legacy (BasicFile) LOB storage and offers significant performance improvements as well as deduplication, compression, and encryption.

- **Security**  Encrypted tablespaces provide additional security and are fairly easy to implement.

As mentioned before, the migration process might be your only chance to make meaningful improvements to your database.

**TIP & TECHNIQUE**
This might be your only opportunity to make improvements. Consider the opportunity and think about how changes might improve your database.

Future Migration, Transformation, and Upgrade Techniques
Finally, this book tackles three of the newest topics in Oracle Database technology. We explore and demonstrate how data migration, data transformation, and database upgrade projects will be accomplished in the brave new world of Oracle Database 12c (Chapter 13), on engineered systems like Oracle’s Exadata hardware line (Chapter 14), and within the realm of Oracle’s Cloud Computing environment (Chapter 15).
Summary

As we have described, there are many ways to perform database migrations. Determining which method is right for your situation depends on your IT organization’s needs and requirements. Remember that this migration might be the only chance that you have to make major structural changes to your database, so be sure to take advantage of this opportunity.

In this book, we provide extensive examples, checklists, and guidance on how to migrate your database from one platform to another. In Chapters 2 and 3, we’ll begin the process of helping you to determine which method is most appropriate for your migration, upgrade, or transformation project, and then we’ll explain how to build exactly the right plan for your impending project.