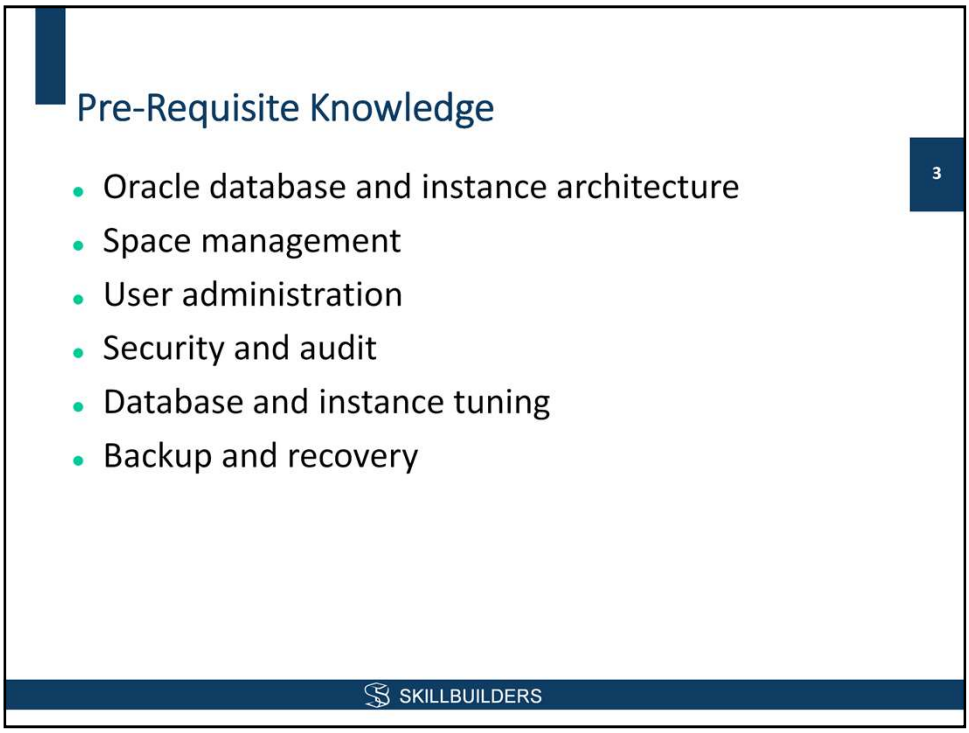
Lesson 1
Introduction

SKILLBUILDERS

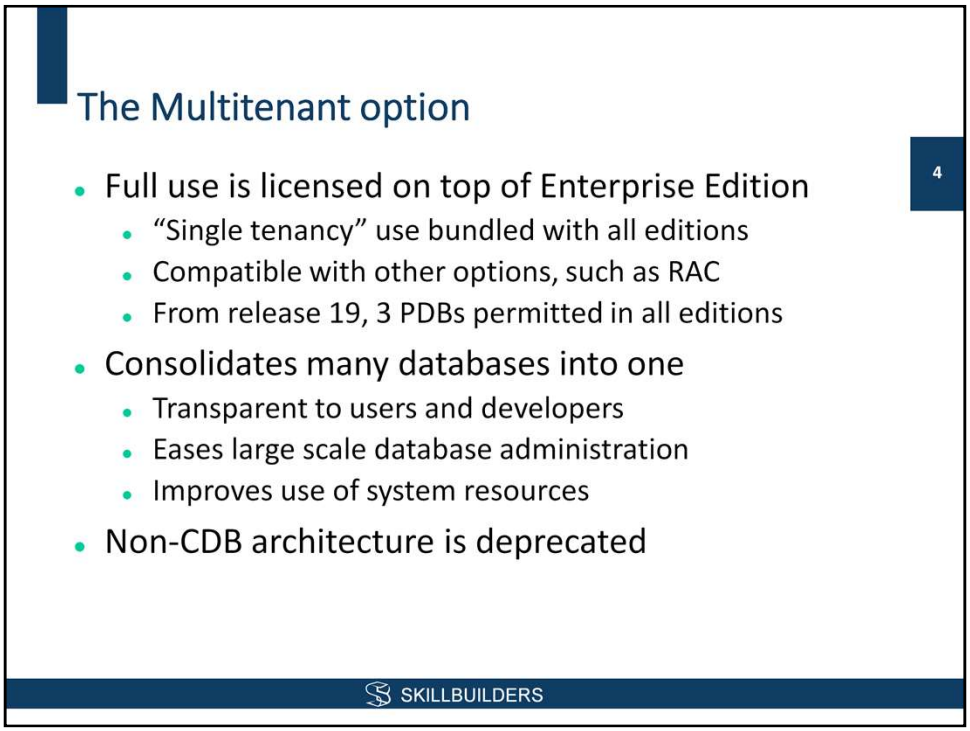
Objectives

- Position the Multitenant option
- Pedagogy
- Connect to the lab environments



Oracle has gone to great lengths to ease the process of migrating to a multitenant environment. Nonetheless, there are some areas of administration where multitenant does add some complexity and the DBA will need to extend his knowledge to administer the environment effectively.

The principles of database and instance architecture, space, user, and security administration, and tuning are the same as in a non-CDB environment but there are huge differences in some of the details.



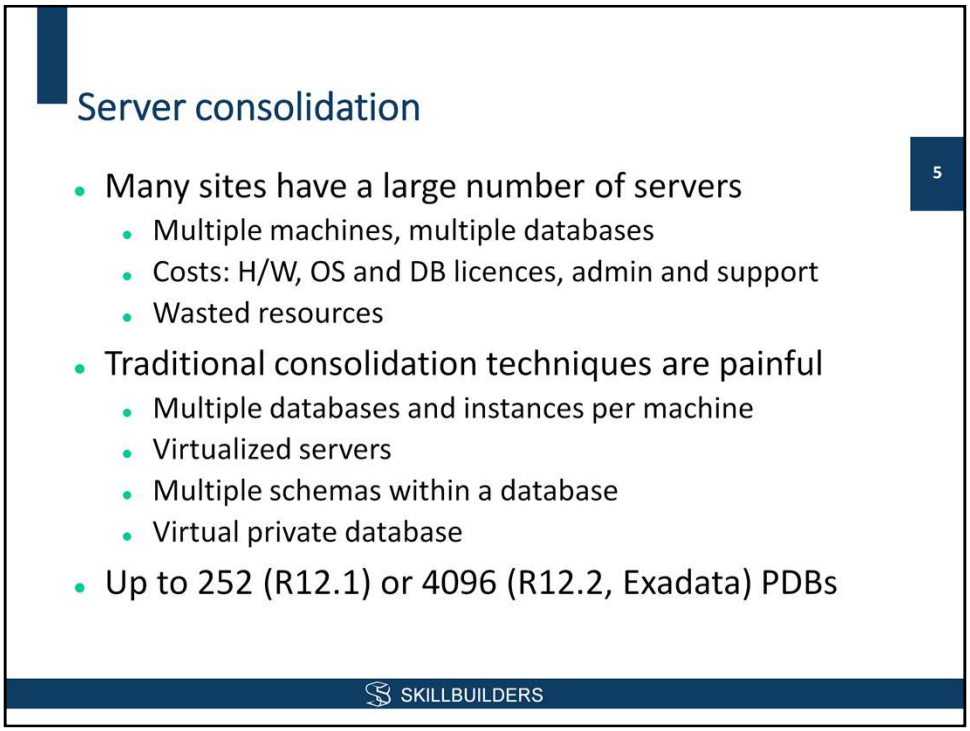
A word on terminology:

Throughout the 12c beta programme, the Multitenant option was referred to as "pluggable databases". When the product went GA, it was re-named and now one is supposed to talk about "pluggable containers". However, the previous term still exists in the names of many views, columns, and procedures. In some cases there are even duplications, such as v$pdbs and v$containers. Traditional databases are now referred to as "non-CDBs" ("CDB" meaning "container database").

There is no option about using Multitenant. The non-CDB architecture was deprecated in release 12.1.0.2. Non-CDB is no longer supported from release 21.

Note that with the change in numbering, release 18c is the 12.2.0.2 patchset. 18c introduces some new features, but nothing dramatic. Release 19c is the 12.2.0.3 patchset, the terminal release of 12c with long term support. 21c is an "innovation" release with only short term premium support and no extended support; 23c is the latest long term support release.

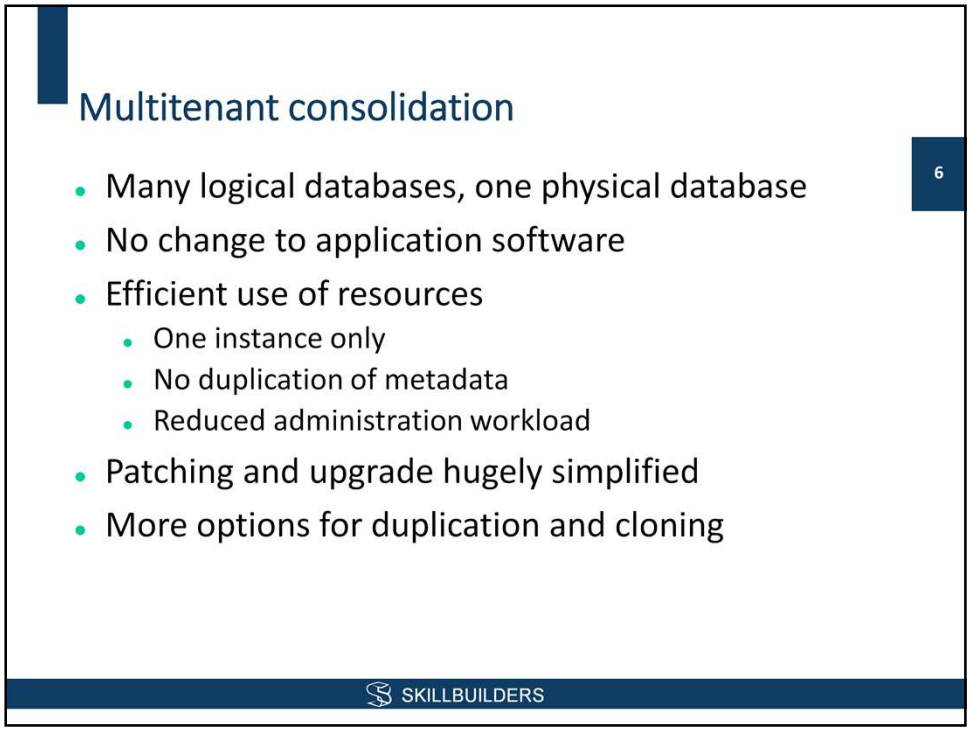
The single tenant model has always been available in SE2 and EE without the Multitenant option. From release 19, the rule was further relaxed and up to three user created pluggable containers are permitted.



The headline usage of Multitenant is server consolidation: a term that (rightly) terrifies many DBAs. However, consolidation is often a business imperative. There can be huge financial savings.

Multitenant is undoubtedly aimed at large scale users. This means cloud service providers, where it opens the possibility of hosting hundreds of clients in one physical database. The maximum is dependent on release and platform: the upper limit of 4096 applies to Exadata in releases 12.2 and higher. The isolation between containers means that no one client should be aware that they have anything other than a dedicated database.

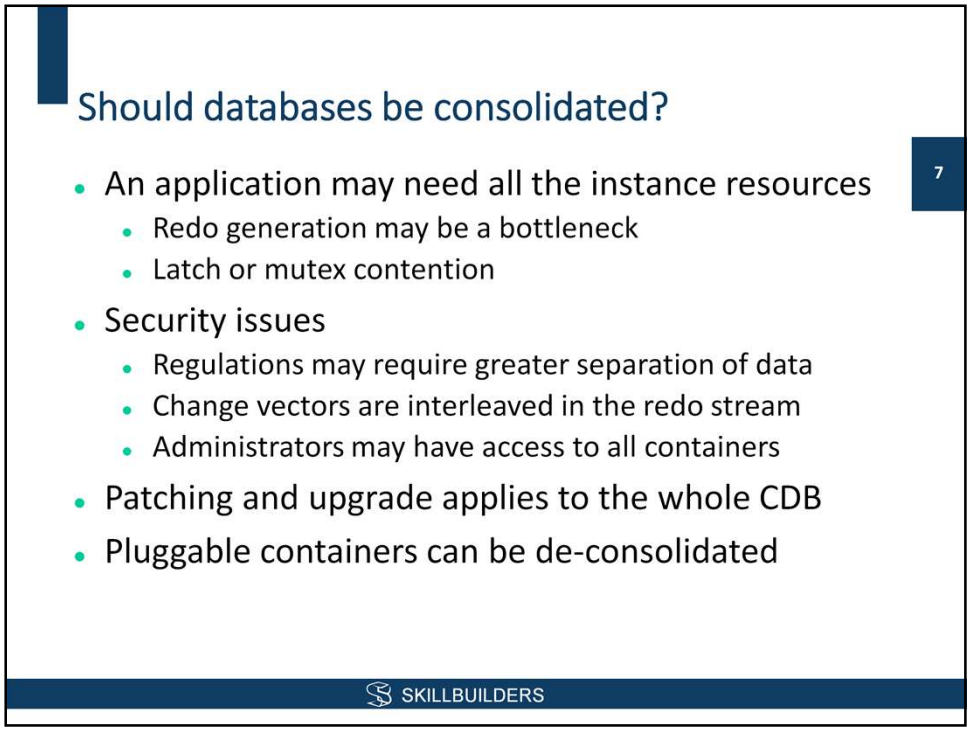
Note that an analogy can be drawn with the way SQL Server operates. A SQL Server "database" is very similar to a pluggable container: an isolated business entity managed by a central shared resource.



Database users (including developers) connect to a pluggable container. This appears to be a normal, self contained, database. The isolation from other containers is perfect, and in virtually all circumstances there is no reason why the users would be aware that they are running in this environment rather than in a non-CDB.

From the SA's point of view, there will be huge efficiency savings. There is one instance rather than many with a consequent saving in background processes. There will be memory savings: the SGA size of a CDB supporting a dozen PDBs need not be sized as the total of a dozen non-CDB's SGAs. There may be some space saving as there is only one copy of all the shared data dictionary objects.

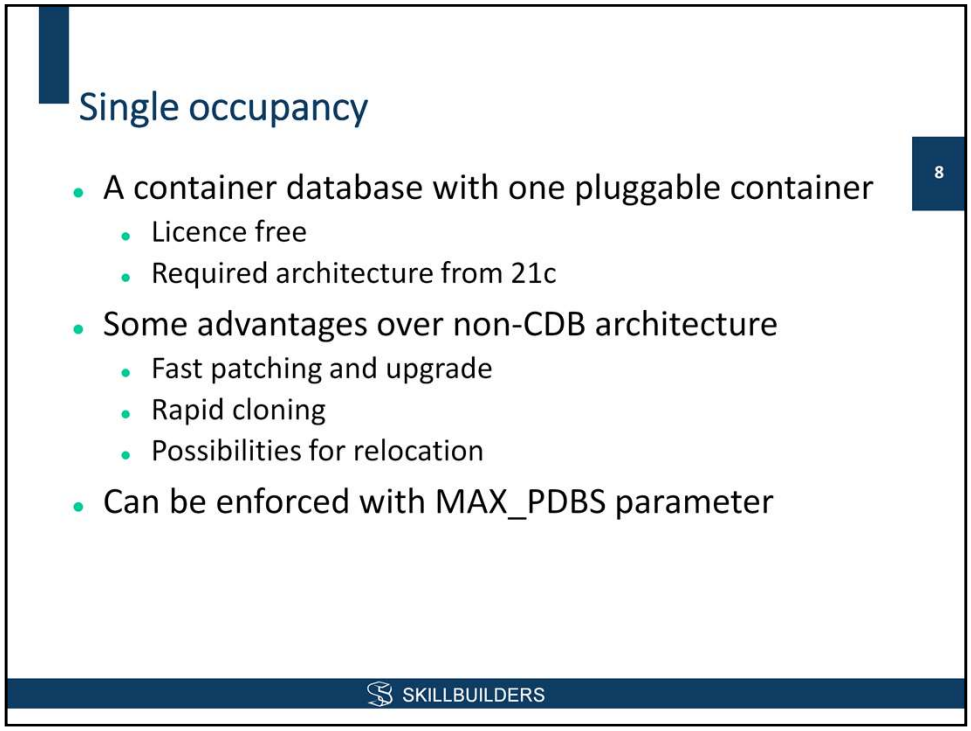
From the DBA's point of view, CDB does add an extra layer of complication, but generally speaking managing one large database is much less work than managing many small ones.



-. There are some points of serialization in the Oracle architecture that multitenant does not mitigate, and if a database is being limited by one or more of these then it is clearly not candidate for moving into this environment.

There may also be security issues. The isolation between pluggable containers is perfect at the application level, but data is intermingled in the SGA, in the redo stream, and possibly the undo tablespace. At the file system level, all the datafiles will be owned by the same operating system user. Whether this is an issue will depend on the application area.

If consolidation has proved to be a mistake, the PDBs can be unplugged and each plugged into its own CDB as a single occupant. It is however impossible to convert a PDB into non-CDB.



"Single occupancy" or "single tenant" means a multitenant container database that has only one pluggable container. This container could be unplugged and either replaced with a different container or plugged back in – but at no time is there more than one pluggable container present.

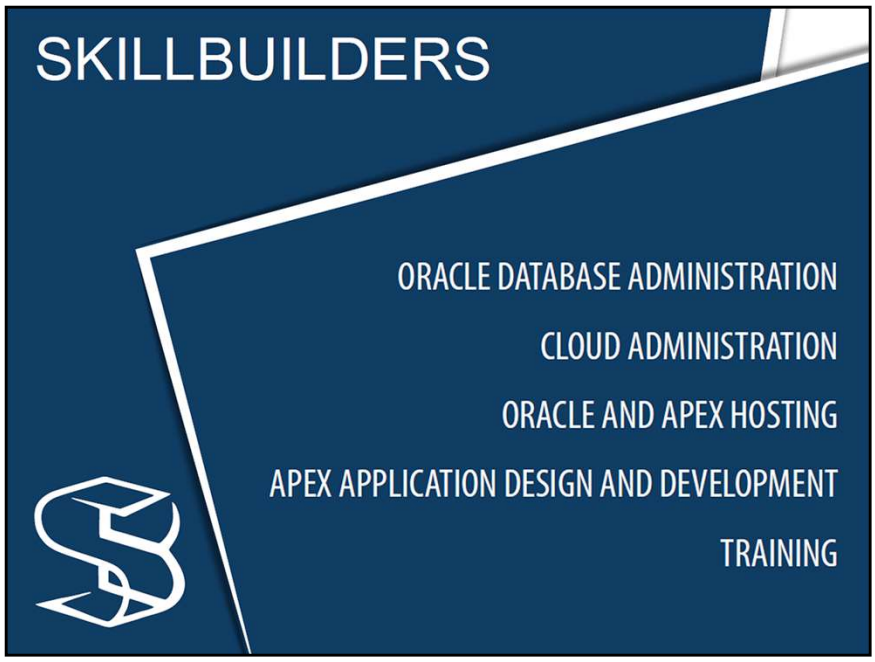
Clearly this model cannot be used for server consolidation but it does give access to the fast and simple facilities that multitenant offers for patching and cloning .

Classroom Environment

- Current release of the database
- Linux or Windows
- Command line and graphical user interfaces
- SQL*Plus, SQL Developer

Introduction: Exercise

- Connect to the classroom servers
  - Graphical interface
  - Command line interface
- Log on to a database with SQL*Plus
- Investigate the environment


SKILLBUILDERS
ORACLE DATABASE ADMINISTRATION
CLOUD ADMINISTRATION
ORACLE AND APEX HOSTING
APEX APPLICATION DESIGN AND DEVELOPMENT
TRAINING