VISCOSITY
PROFESSIONAL SERVICES

Prepare for Oracle Database 23c

Prepare to Upgrade to Oracle Database 23c
Sprinkles of Oracle Database 23c New Features

New York Oracle User Group
May 2023

Charles Kim

Oracle ACE Director | CEO

@racdba
linkedin.com/in/chkim/
ckim@viscosity.com

Charles Kim

Oracle ACE Director

- Oracle ACE Director
- Founder and CEO of Viscosity
- Author of **11 Books** in the Oracle Ecosystem
- Oracle Expertise: Mission Critical Databases:

  Exadata, RAC, Data Guard, ASM, RMAN, Shareplex/GoldenGate
- Specialize in "**Complex Data Replication** & **Integration"** with **Shareplex** & **GG**
- Oracle Management Cloud Certified

- Oracle Exadata Certified Implementation Specialist
- Oracle Certified RAC Expert
- Oracle Linux Certified Expert
- Former President of the Oracle Cloud SIG for several years

ORACLE Platinum Partner | Quest PLATINUM+ PARTNER | Microsoft Gold Partner | IN-COR-TA | vmware vEXPERT | CERTIFIED vmware PROFESSIONAL | redhat CERTIFIED | ORACLE Certified Professional

viscosityna.com

@ViscosityNA

Viscosity's Oracle ACEs

The Oracle ACE Program

The Oracle ACE Program recognizes and rewards individuals for their contributions to the Oracle community.

Charles Kim
CEO | Co-Founder

@racdba

ACE Director

Rich Niemiec
Chief Innovation Officer

@richniemiec

ACE Director

Craig Shallahamer
Applied AI Scientist

@orapub

ACE Director

Sean Scott
Principal Consultant

@oraclesean

ACE Director

Gary Gordhamer
Principal Consultant

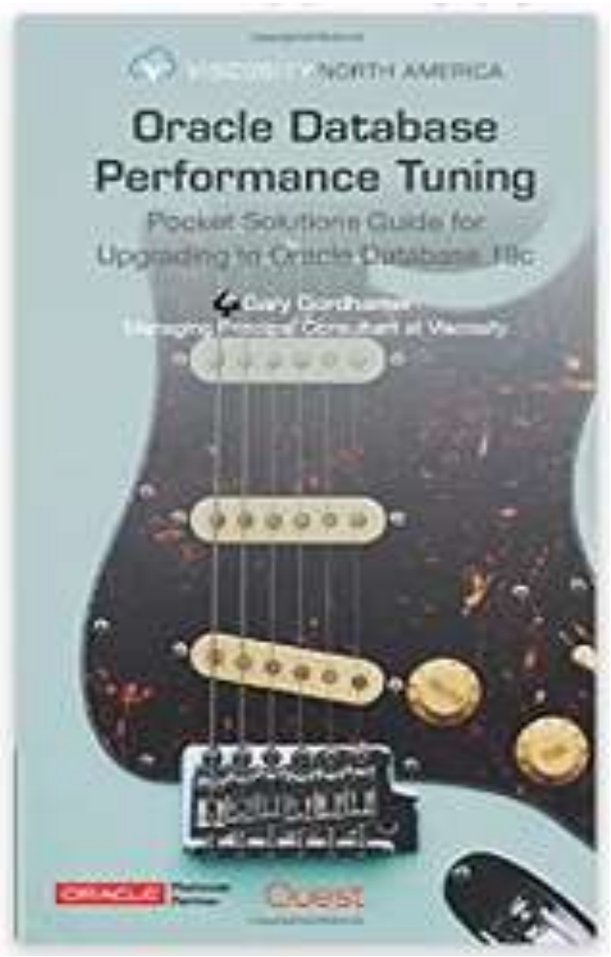
@ggordham

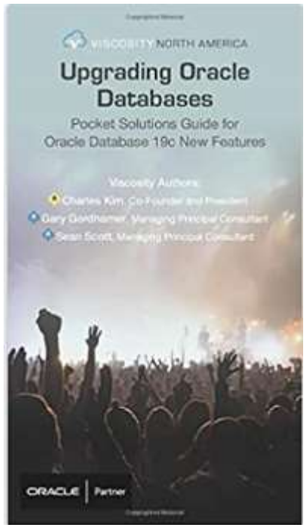
ACE Pro

Julio Ayapan
Principal Consultant
Oracle ACE
Alumni

viscosityna.com

@ViscosityNA





Get Your Own Signed Copy

Upgrading to Oracle Database 23c

Updating both books for Oracle Database 23c

Big Focus on AutoUpgrade

Performance Tuning

Additional 23c New Features

ORACLE®

DATABASE APPLIANCE

Webinar: Innovations in ODAs

12:00 EST
SOUG.ORG. May 25th:

Oracle Database Appliance (ODA) is Oracle's entry level engineered systems product designed by Oracle's database group. It's designed to be simple, optimized to run databases, and affordable. The **ODA** has many features and capabilities that are not available on any other platform.

In this webcast, we will go over **Oracle Database Appliance**'s latest updates and recent innovations that makes the **ODA** a great platform or Oracle customers for both databases and applications. These enhancements improve the performance, scalability, and security of **ODA**. Join us to learn how these new features and capabilities can benefit your organization.

Evolution of the Oracle DBA

Kind of DBA	Timeline
CLI DBA	Early 90's DBAs
GUI DBA (Right Click DBAs)	Late 90's and Dot Com
Google DBA (Developers)	Dot Com and 2000's
iDBA	Dot Com, IOUG iDBA Master Curriculum
RAC DBAs (MAA DBAs)	2000+ after 9.2 (but major spike with 10.2) **+ Data Guard**
DMA	2010+ Database Machine Administrator
vDBA / vRAC DBA	2010+ Evolving role of a DBA in the virtual world
Cloud DBA	2011+ Database Consolidation with Private Database Cloud Oracle Database 12c Launches June 2013
Public Cloud DBA	*2015+ Oracle Public Cloud with Database Cloud Service, Database Backup Cloud Service, Storage Cloud Service, IaaS Cloud Service*
PDBAs	2017+ Multi-Tenant with Oracle Database 12c Release 2 GA – March 2017
Oracle 18c	February 2018 in Oracle Cloud, July 2018 On-Premise
Oracle 19c	**January 2019 - LiveSQL** **February 2019 for on-premises Exadata; April 2019 General Release (Terminal Release or Long Term Release)**
Oracle 20c	**Dead! Made preview in the cloud b[illegible]**
Oracle 21c	**January 2021 - Preview released in C[illegible]ase** **August 2021 – GA On-Premise**
Oracle 22c	**Skipped**
Oracle 23c	**Next Long Term Release**

Converged DBA

"It is not the strongest or the most intelligent who will survive but those who can best manage [adapt to] change."

The DBA role is not dying.

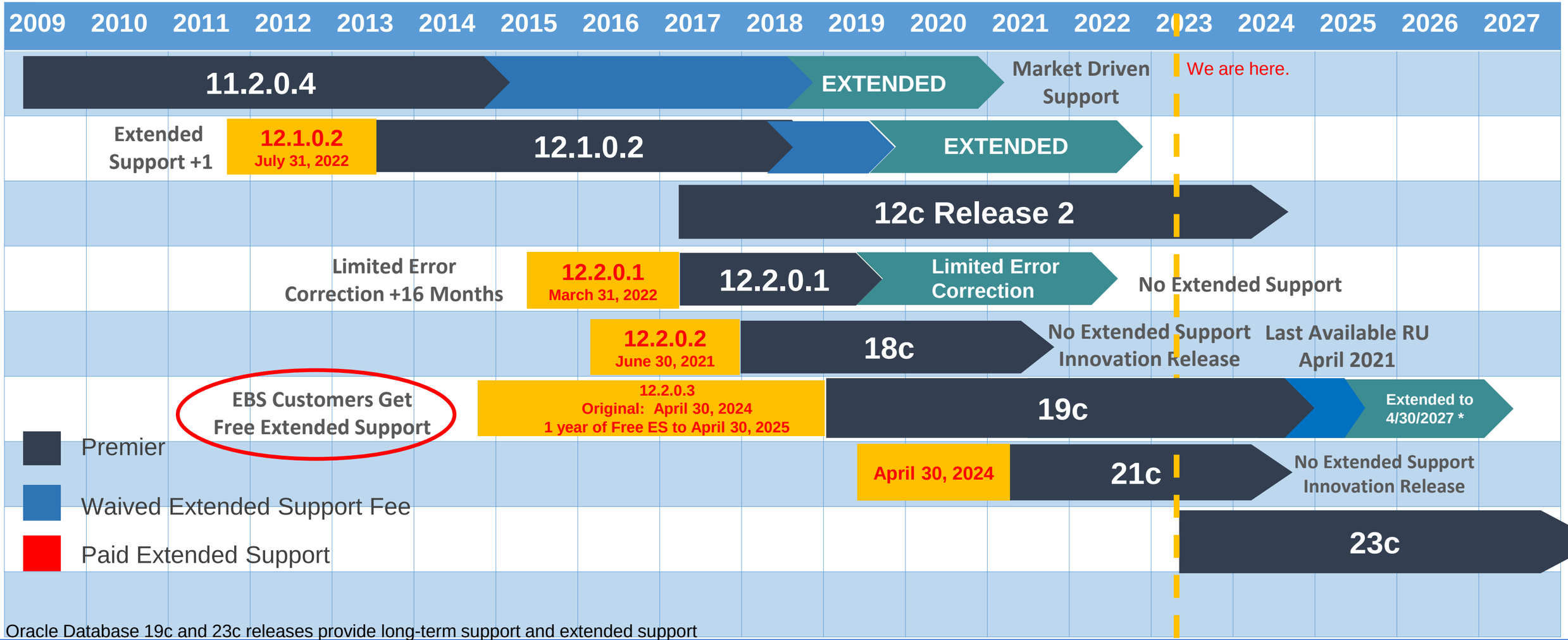
The DBA role has always been changing.

Oracle Release Dates

Oracle Database 11g Release 1	Aug 2007
Oracle Database 11g Release 2	Sept 2009
Oracle Database 12c Release 1	June 2013
Oracle Database 12c Release 1 (Patchset) - 12.1.0.2	June 2014
Oracle Database 12c Release 2 (Cloud)	September 2016
Oracle Database 12c Release 2 (On-premise)	March 2017
Oracle 18c - “Cloud First”	February 2018
Oracle 18c – Available on Exadata	February 2018
Oracle 18c – Available on ODA	March 2018
Oracle 18c – On-Premise	July 2018
Oracle 19c - Exadata On-Premise Oracle 19c - General Availability - Terminal Release	Feb 13, 2019 April 2019
~~Oracle Database 20c Preview Release on OCI Only (R.I.P)~~ 	Feb 14, 2020
Oracle Database 21c on OCI (Cloud First)	December 2020
Oracle Databse 21c – On-Premise	August 2021
Oracle Database 21c XE Edition	September 2021
Oracle Database 23c Developer Release	April 3, 2023
Oracle Database 23c EE GA	Q2? 2023

Lifetime Support Commitments and Plans

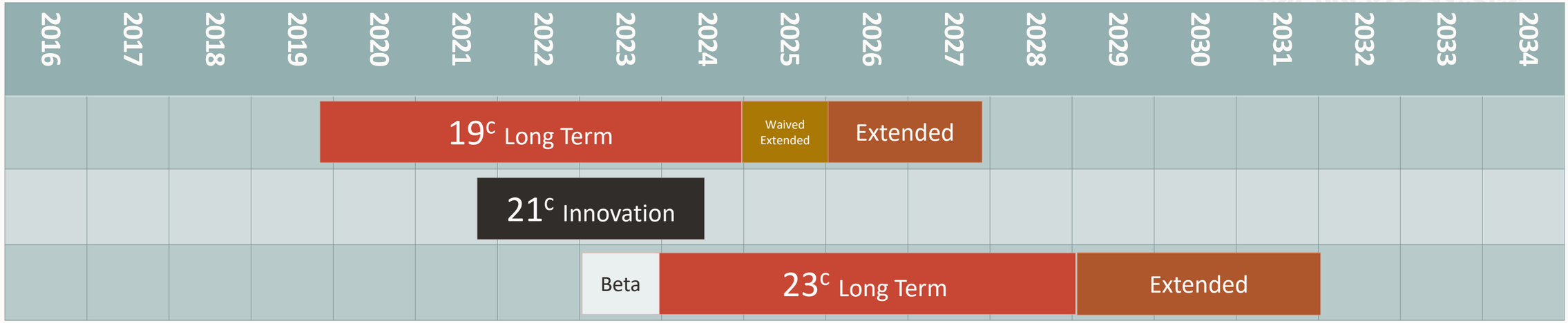
Planned Release and Support Timelines - Subject To Change



Oracle Database 19c and 23c releases provide long-term support and extended support

- Innovation Release - 2 years of Premier Support, and no Extended Support
- Long Term Release - 5 years of Premier Support, and 3 years of Extended Support
- Check MOS Note **742060.1** for the latest schedule

Projected Database Release and Support Timeline



- Innovation Release - 2 years of Premier Support, and no Extended Support
- Long Term Release - 5 years of Premier Support, and 3 years of Extended Support
- Always refer to MOS Note: Release Schedule of Current Database Releases (Doc ID 742060.1)

Top Reasons Why Customers Upgrade To Oracle Database 23c

New and Improved Database Features	• **Security** • Converged Database Enhancements • **Performance** • Higher Availability
Compliance and Regulations	
Long-Term Support	• Extended Support
Attract Talented People	• No one wants to work for a shop that is a dinosaur

What is Your Timeline to Upgrade to 23c

Database Upgrade Considerations

Top 10 Questions to ask yourselves:

1. **How long does it take** for your company to upgrade? DEV/QA/UAT/PRODHow many databases do you have? How many environments do you have?
2. Have you already adopted **PDBs**? Do you have a **complicated** environment? RAC? Exadata?

3. What are the **key new** features that you will adopt as part of the upgrade?
4. Are you **migrating** to the cloud as part of the upgrade? Or are you upgrading **hardware** or OS as part of the upgrade? Are you **modernizing** your stack?

5. What were the **biggest pain points** from you last upgrade? How can we mitigate against them?

What is Your Timeline to Upgrade to 23c

Top 10 Questions to ask yourselves:

6. Will **applications** need to be **updated**? Do we need to factor version compatibility with the new database version?
7. Will ODBC and JDBC **drivers** need to be updated on the clients, ETL servers, application servers?

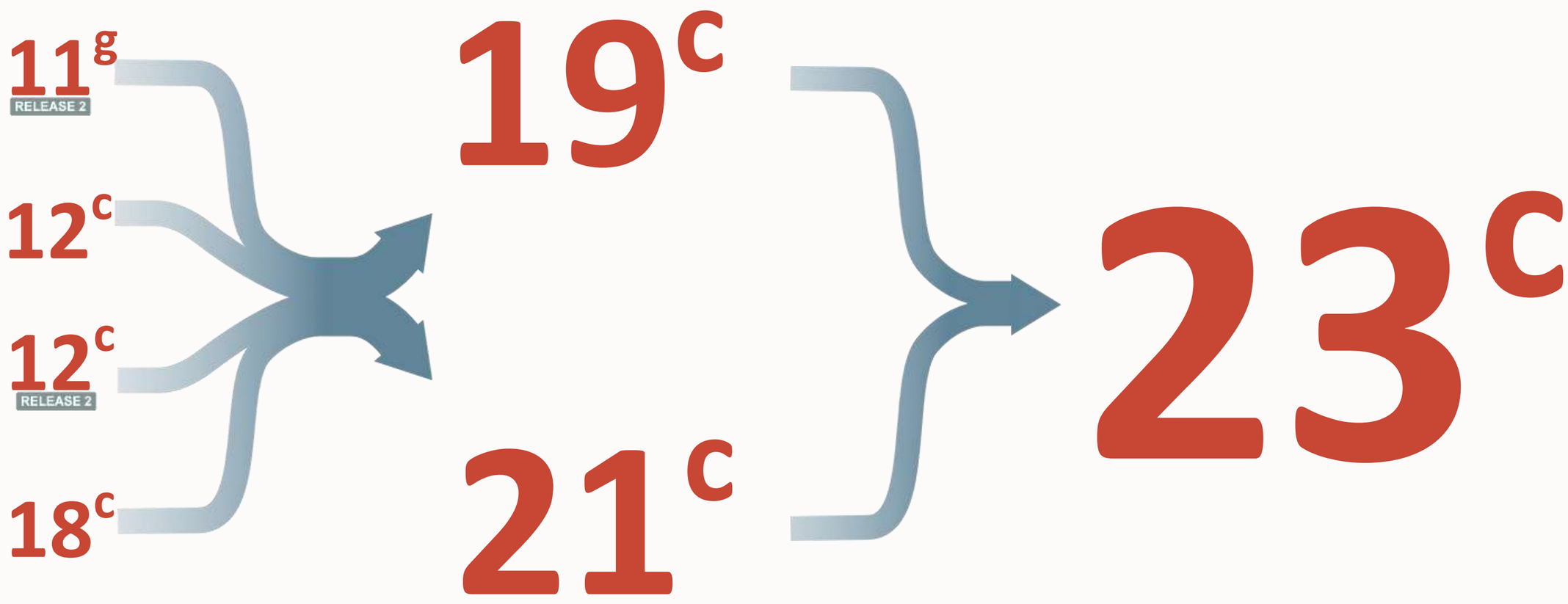
8. Do you have **database links** to other databases? Do those databases need to be upgraded? Is there compatibility issues? Are you using **heterogeneous** services to SQL Server?
9. Does your organization have the **skillset** to support the upgrade efforts? Does your team have the skillset to support the database (more so the application teams) after the upgrade?

10. Do you have a **test** plan? Do you have **backout** plans?
With compliance and QA teams, how fast can you push the upgrade?

How do we get to 23c?

Oracle database upgrade paths

Upgrade Path to Oracle Database 23c

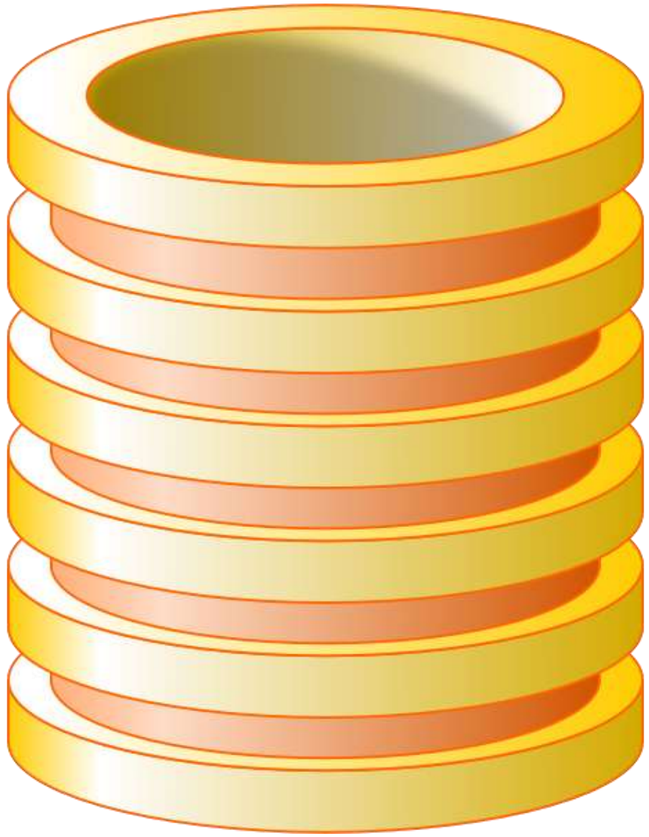


Upgrade Path to Oracle Database 23c

Oracle Database Release	Must Upgrade to Interim Jump	Target Version
11.2.0.4	Oracle Database 19c or Oracle Database 21c	23c
12.1.0.2 (12c Release 1)		
12.2.01 (12c Release 2)		
Oracle Database 18c		

The Compatible Initialization Parameter

Oracle Database Release	Default Value	Minimum Value
Oracle Database 23c	23.0.0	19.0.0
Oracle Database 21c	21.0.0	12.2.0
Oracle Database 19c	19.0.0	11.2.0
Oracle Database 18c		11.2.0
Oracle Database 12c Release 2 (12.2)		11.2.0

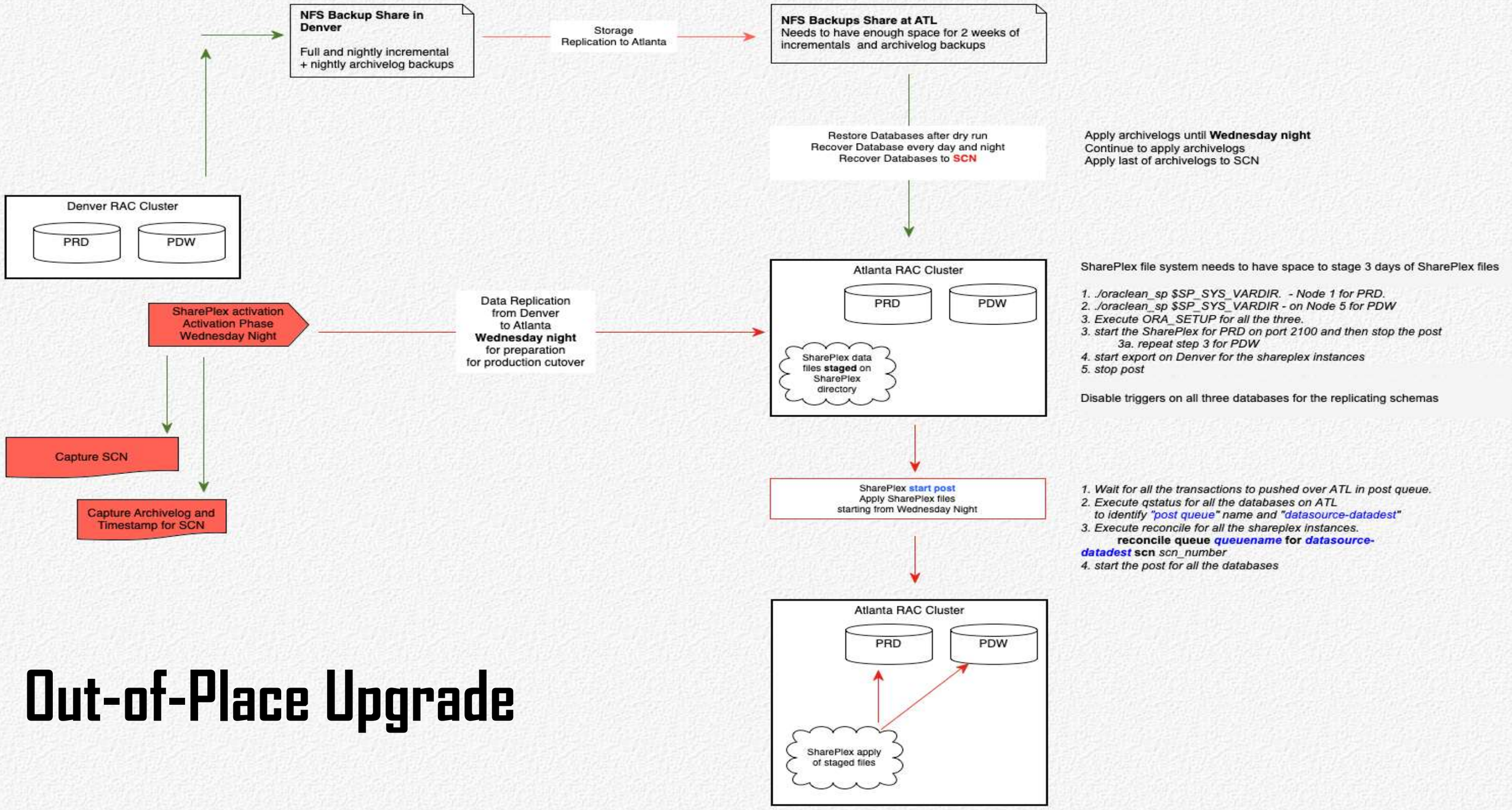


Upgrade Methods

Method	Upgrade From	Upgrade To
Datapump Export/Import	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c	23c
AutoUpgrade DBUA DBUpgrade	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c	19c Then to 23c
	19c	23c
Transportable Tablespace	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c	23c
GoldenGate / SharePlex	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c	23c
Transient Logical Approach DBMS_ROLLING	11.2.0.4, 12.1.0.2, 12.2.0.1, 18c	19c Then to 23c
	19c	23c

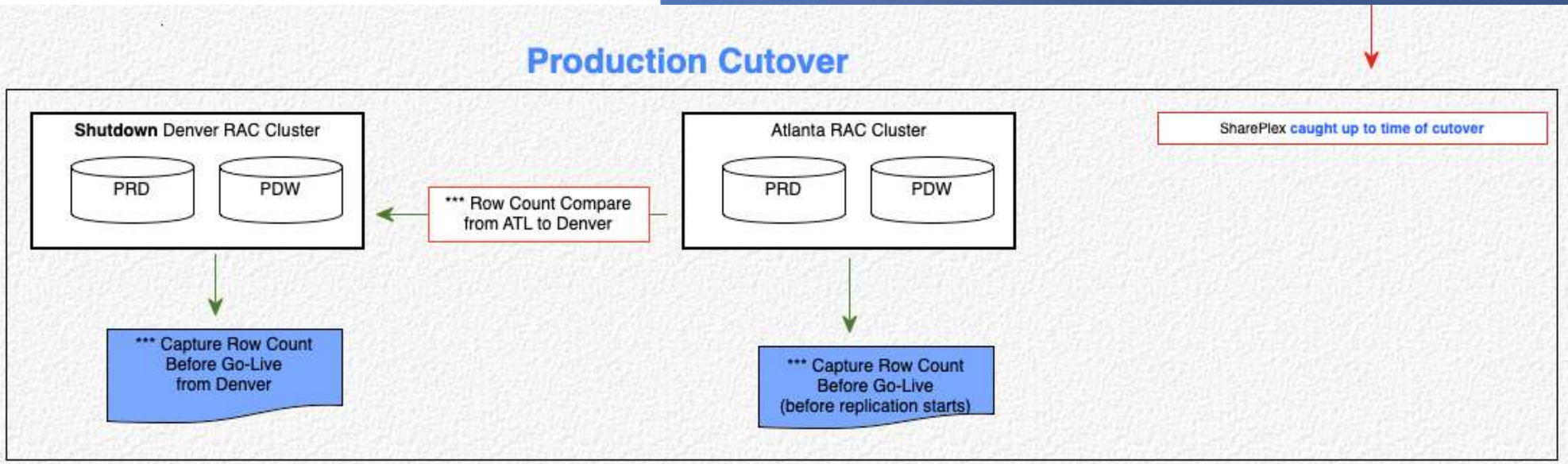
Near-Zero downtime upgrade

- Leveraging products like Oracle GoldenGate or Quest SharePlex, we can help customers perform zero-downtime upgrades to Oracle Database 23c
- By incorporating reverse replication back to the source database, create an insurance policy for the company so that you can perform a zero-risk database upgrade value proposition



Out-of-Place Upgrade

Production Cutover Row Count Validation



SharePlex caught up to time of cutover

Friday night
1) Break connection from Phoenix - stop SharePlex
2) Enable Triggers

Oracle 19c - Oracle's Recommended Roadmap

- Oracle **strongly recommends** that customers upgrade their databases to Oracle Database 19c
  - 19c is the terminal release of Oracle Database 12c Release 2
    - 19c is the **long-term support** or extended support will only be offered on this version
      - **Oracle Database 19c will have premier support up to ~~March 31, 2023~~. April 30, 2024.**
      - **Extended support will be provided until ~~March 31, 2026~~. April 30, 2027**

- Oracle Database 23c will go **GA** this year but most customers will not venture there until next year

MRPs for Linux Only

Monthly Recommended Patches (MRP)
Start on November 2022
Linux Platforms Only

- To provide more frequent access to recommended and well-tested collections of patches

- Starting on October 2022, 19c RURs will no longer be provided for 19.17.0 and above

Notes:

- Sunsetting of 19c RURs and FAQ (Doc ID 2898381.1)

“ MRPs may include critical 3rd Party Security vulnerabilities. The Quarterly Database Release Update (RU) will continue to be primary mechanism for delivery of Database Security vulnerabilities announced as part of the Quarterly Security CPU program”

Oracle Database 19c

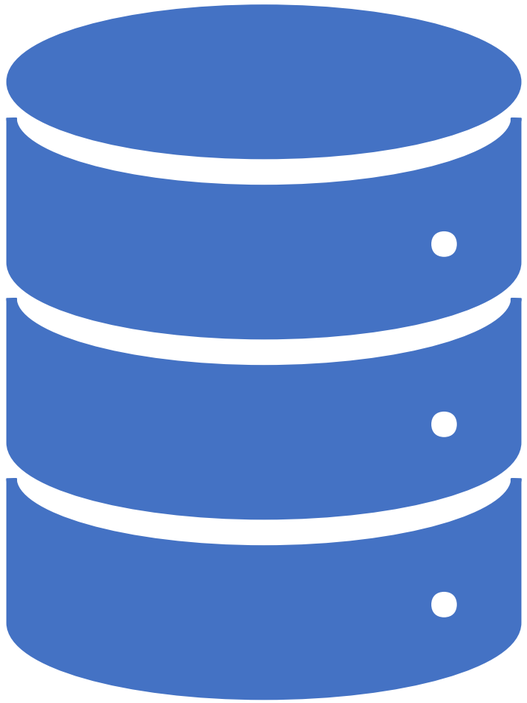
Upgrade Checklists and Patch Information

Best Practice

N-1

555.1	**Oracle Database 19c Important Recommended One-off Patches (Doc ID 555.1)** **Death of the RURs ☺** **Welcome MRPs**
2521164.1	**Oracle Database 19c Proactive Patch Information** **Start Here For Patch Downloads**
2539778.1	Oracle 19c - Complete Checklist for Manual Upgrades to Non-CDB Oracle Database 19c
2543981.1	Oracle 19c - Complete Checklist for upgrading Oracle 12c, 18c Container Database (CDB) to Oracle 19c Release using DBUA
2549866.1	Oracle 19c - Complete checklist for Manual Upgrade for upgrading Oracle 12.x, 18c Container database (CDB) to Oracle 19c (19.x)
2545064.1	Oracle 19c - Complete Checklist for Upgrading to Oracle Database 19c (19.x) using DBUA
2539778.1	Oracle 19c - Complete Checklist for Manual Upgrades to Non-CDB Oracle Database 19c
2539751.1	Patches to apply before upgrading Oracle GI and DB to 19c
2542082.1	19c Grid Infrastructure and Database Upgrade steps for Exadata Database Machine running on Oracle Linux
1587357.1	Oracle Database (RDBMS) on Unix AIX,HP-UX,Linux,Solaris and MS Windows Operating Systems Installation and Configuration Requirements Quick Reference (12.1/12.2/18c/19c)

PDB



Oracle Database 21c - The first CDB-Only Release

PDB is THE FUTURE: Changes How DBAs Work

Oracle Database 23c is right around the corner

Oracle Database19c

3 PDBs - FREE TO USE

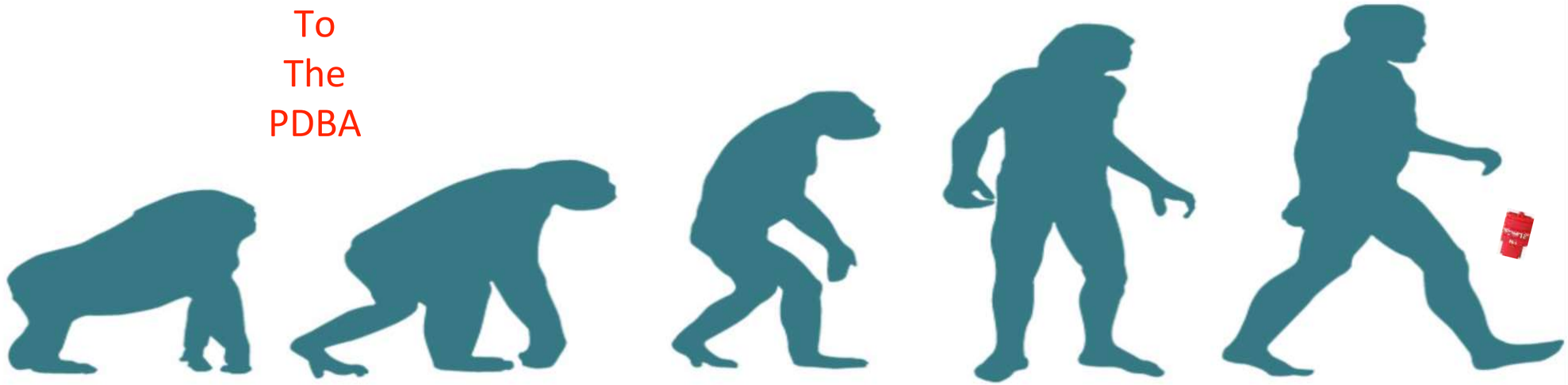
Starting in Oracle Database 19c, each container database can have up to 3 pluggable databases

- Change how we perform database **upgrades**
- Change how we do data center **migrations**
- Change how we perform **patches**

19c ORACLE Database

Evolution of Multitenant (PDB) Features

Evolution of the
Dinosaur DBA
To
The
PDBA



Read Only Source – **Cold/Remote Clone**

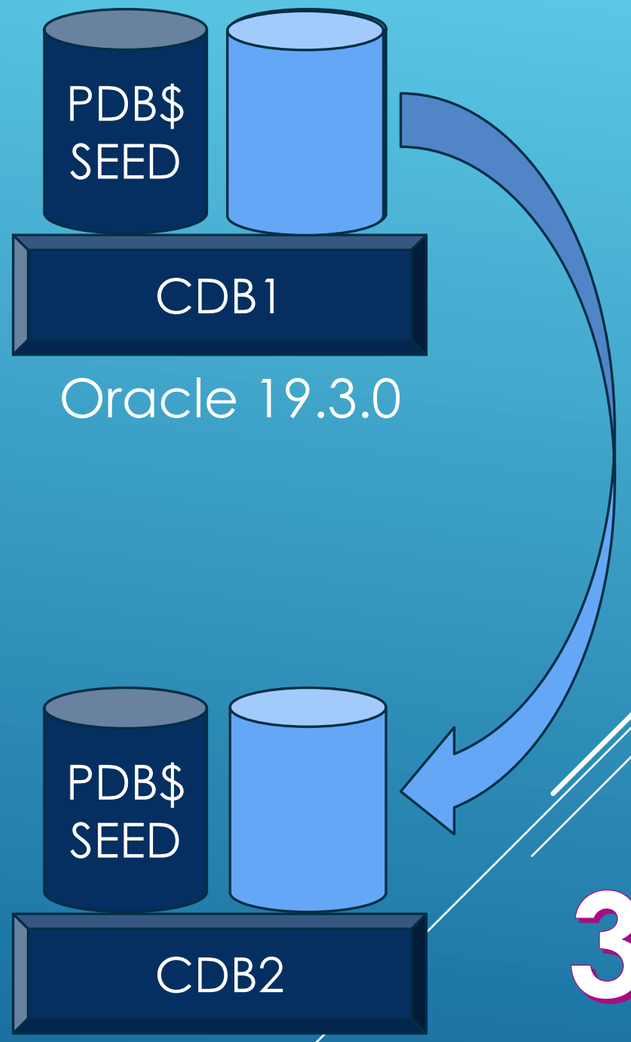
Read Write Source – **Hot/Refresh Clone**

Online **Relocate**

NEW IN 12.2

- **Lockdown Profiles**
- **PDB Archives**
- **Proxy PDBs**
- **Refreshable PDB Switchover (18c)**
- **Refreshable PDBs (12.2)**
- **Snapshot Carousels (18c Exa)**

PATCHING

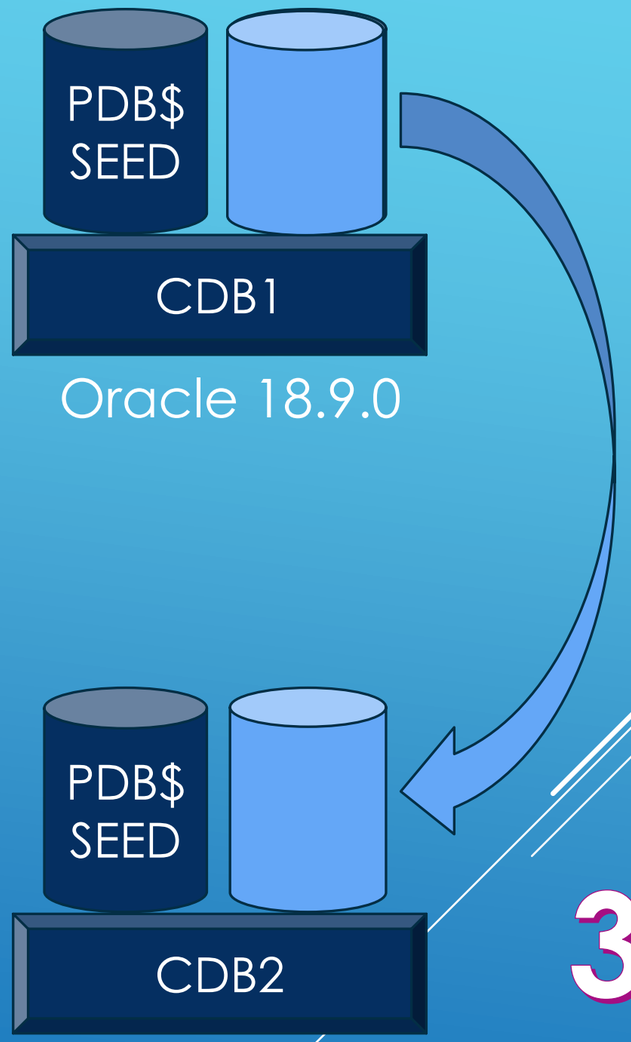


1 Create new CDB2 with same components as CBD1 - make sure COMPATIBLE is identical

2 Relocate PDB

3 Execute ‘datapatch –verbose’

UPGRADING



1 Create new CDB2 with same components as CBD1 – This time as an Oracle Database 19c

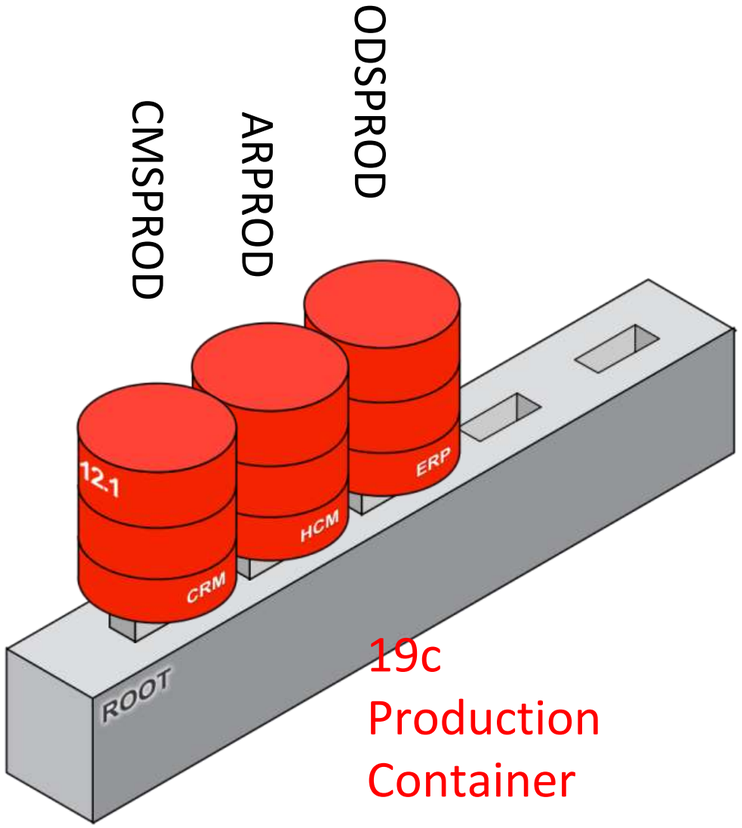
2 Clone PDB

3

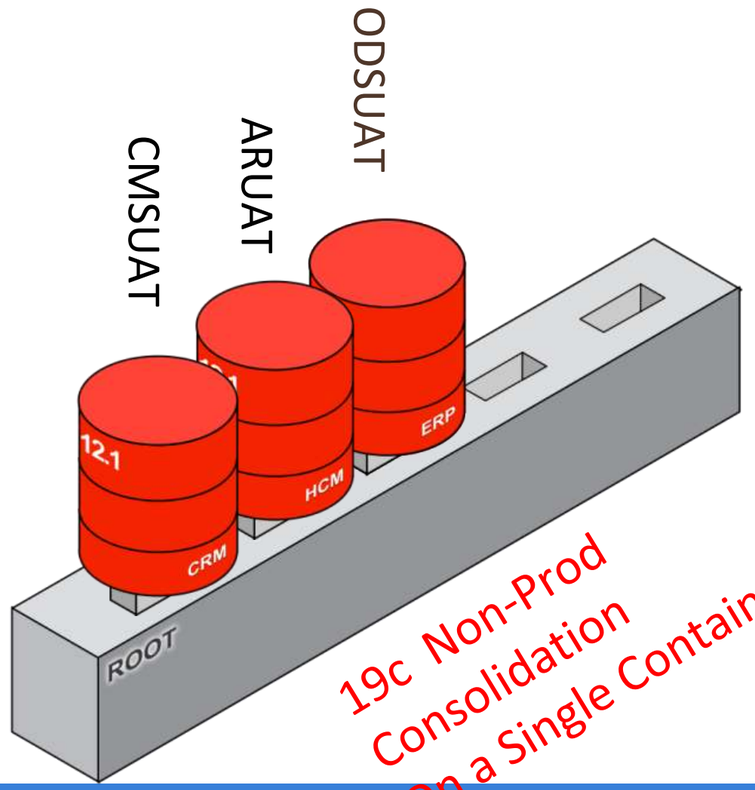
```
SQL>  alter pluggable database open upgrade;

$ perl $ORACLE_HOME/rdbms/admin/catctl.pl -d \
$ORACLE_HOME/rdbms/admin -c 'PDB' \
-l $ORACLE_BASE catupgrd.sql
```

19c PDB Consolidation for Cost Containment
Standardize in consolidating multiple databases (customers) into a single container



19c
Production
Container



19c Non-Prod
Consolidation
On a Single Container

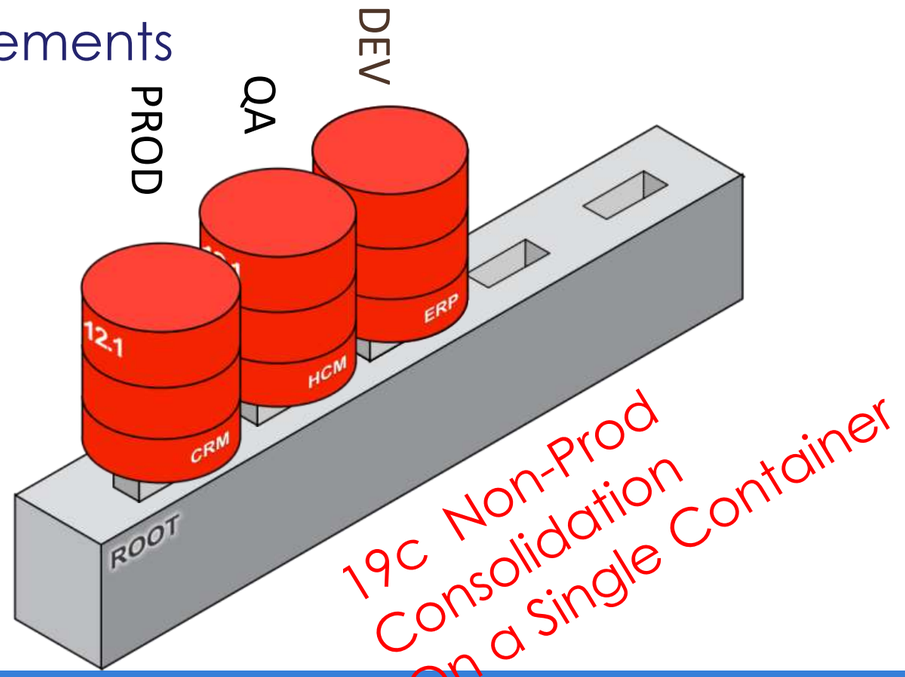
PDB Complete Isolation
Meets Compliance & Regulatory Requirements

19c PDB Consolidation for Cost Containment
Consolidate databases into a single container

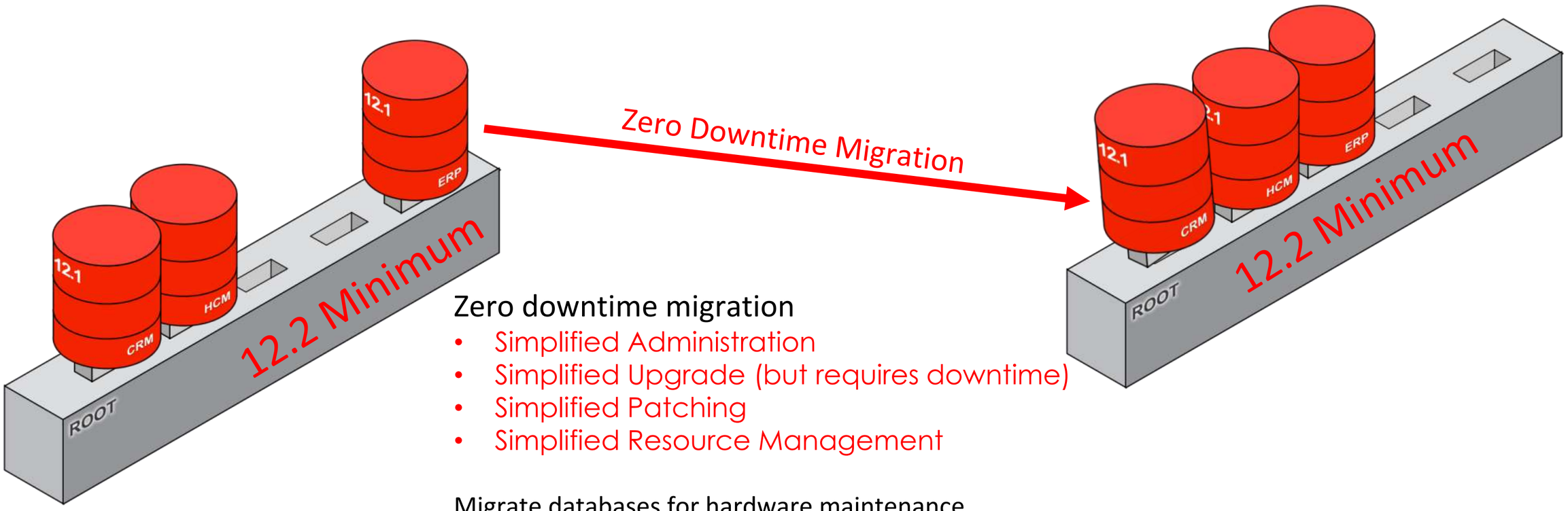
Not a recommended deployment model

Can put PROD, QA and DEV on a single container

PDB Complete Isolation
Meets Compliance & Regulatory Requirements



19c PDB Database
High Availability



Zero downtime migration

- Simplified Administration
- Simplified Upgrade (but requires downtime)
- Simplified Patching
- Simplified Resource Management

Migrate databases for hardware maintenance
Migrate databases for OS upgrades

More PDB

Dynamic CPU Scaling

CPU Min Count at PDB Level

- Works with Oracle Database Resource Manager (DBRM)
- Starting in Oracle 19.4, CPU_MIN_COUNT is available to set **minimum** CPUs when the system in under load conditions
  - Reserve CPU
  - Protect from noisy neighbors
- Set in conjunction with CPU_COUNT to set the **upper** limit

Fined Grained PDB Patching

PDB is the Future

- Patch individual PDBs in a Multi-tenant environment
- Allows bug fixes to be patched only on specific PDBs rather than across the entire CDB

High Availability Considerations

Active Data Guard: DML Redirection

DBMS_ROLLING with ADG: Introduced in 12.1.0.1
Transient Standby: Started with 11.1.0.6

23c: Flashback Time Travel

Phase 1: High Availability Manager (H.A.M.)

- Target Date: July 31, 2023
- **Version: V1 in the Azure Market Place**
- Initial release will come with 2 days of consulting

Solution: Viscosity will release a High Availability Solution (HAS) for Oracle Databases to address disaster recovery as well as localized failover to replace RAC for HA. For the initial release, Viscosity will offer 2 days of professional consulting to help customers instantiate high availability for their primary database. The customer will have access to a rich UI application to help manage and monitor a Data Guard environment. Our solution does not need Oracle Enterprise Manager for Data Guard management.

V1 features and functionality:

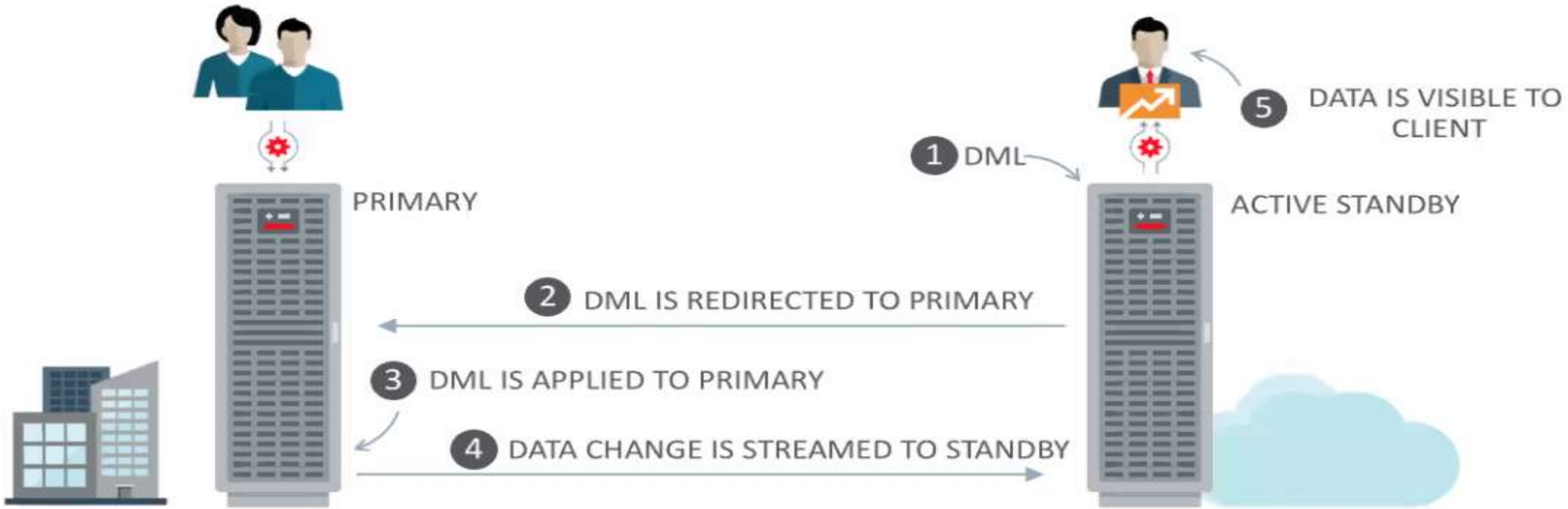
- Instantiate Data Guard with the Broker
- Configure FSFO
- Web Tier for application UI
  - WebUI will launch process to instantiate a physical standby
  - WebUI will launch setup the DG Broker
  - WebUI will setup FSFO with the Observer Process
  - The WebUI will run on the same server as the Observer Processes
- Database server agents in Python to execute server-side scripts
  - The Agent listener will run on the primary database
  - The Agent listener will run on the standby database
  - Returns output back to the web application
- UI will have screens for transport and apply lag monitoring
- UI will have screens for archive gaps
- UI will have screens for Data Guard best practice checks
  - Check for standby redo logs
  - Check for corruption in the datafiles
  - Check for redo generation rate
  - Check for redo volume rate

Read Mostly Physical Standby

19c ORACLE Database

Active Data Guard : DML Redirection

- Read Mostly Standby



19c: Data Guard DML Redirection

RUN DML on Active Data Guard

- Re-Direct DML statement back to the primary database
- Data changes will happen on the primary database
- Changed blocks will ship to ADG
- ADG wil be in sync to maintain redundancy

- Set ADG_REDIRECT_DML **initialization parameter** = TRUE
- SQL> ALTER **SESSION** ENABLE ADG_REDIRECT_DML;

Observer and FSFO

Observer in Observe Only Mode

Configure it to Observe only creating a '**test mode**' to see when a failover or other interaction would have occurred during the normal production processing

- Can tune FSFO for finer granularity
- Can see IF FSFO would have actually occurred without production impact
- Improve failover validation

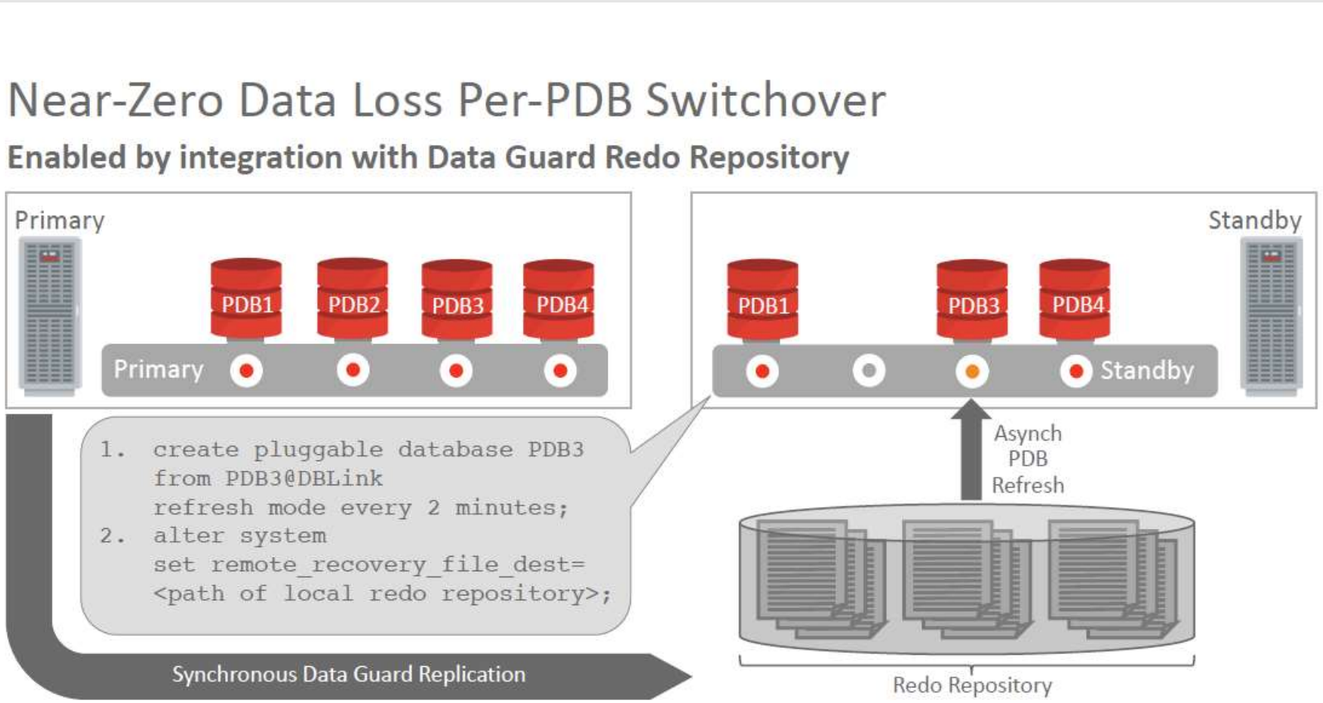
Dynamically Change FSFO target	With 12.2, we can have multiple FSFO targets With 19c, we can change the targets without FSFO being disabled

Additional 19c New Features

Clear Flashback logs periodically to increase FRA size predictability	Starting in 19c, the management of space in the fast recovery area is simplified Oracle Database monitors flashback logs in the fast recovery area and automatically deletes flashback logs that are beyond the retention period. When the retention target is reduced, flashback logs that are beyond the retention period are deleted immediately.
New Parameters for Tuning Automatic Outage Resolution	DATA_GUARD_MAX_IO_TIME sets the maximum number of seconds that can elapse before a process is considered hung while performing a regular I/O operation in an Oracle Data Guard environment. Regular I/O operations include read, write, and status operations. DATA_GUARD_MAX_LONGIO_TIME sets the maximum number of seconds that can elapse before a process is considered hung while performing a long I/O operation in an Oracle Data Guard environment. Long I/O operations include open and close operations.

23c Data Guard & HA Enhancements

Remote Data Guard Redo Repository (18c)



- **A Component of the Data Guard Far Sync**
- Requires ADG license

Advantages

- Minimizes impact of scanning redo logs
- Minimizes amount of redo transport between CDBs
- Enables **near-zero data loss architecture**

RAC

Oracle RAC on Kubernetes & Podman

Dual Instance Rolling Patching

Per-PDB Data Guard Integration Enhancements

Reconnect Refreshable PDBs to source (23c)

Disconnect, Open PDBs as Read-Write and then reconnect the PDB as a refreshable PDB

Essentially like a snapshot standby database

Per-PDB Time Zone (23c on-premise) for Sysdate queries

PDB-level Data Guard (21c)

CDB1:PDB1: Primary --> CDB2: PDB1 DR: Standby

CDB2:PDB2: Primary --> CDB1:PDB2 DR: Standby

Where CDB1 is the primary for PDB1 and CDB2 is the primary for PDB2

Read-Only Per PDB Standbys (23c)

Core Database

Increased number of columns to 4096

Managing Flashback Database Logs Outside the Fast Recovery Area

..

..

..

RUR's are transitioning to MRPs (available on Linux x86-64)

Monthly Recommended Patches (MRPs)

Flashback Time Travel Enhancements

Create Flashback Archives in Your Database

- Flashback Time Travel helps to meet compliance requirements
- Track and archive transactional changes to tables including schemas
- Enable tracking of DML (such as INSERT and DELETE)
- Enable tracking of DDL operations on tables (such as creating and truncating tables)
  - Archive the changes made to the rows of the table in history tables
- Flashback Time Travel maintains a history of the evolution of a table and schema.
  - Enables you to issue flashback queries (AS OF and VERSIONS) on the table and its schema.
- You can view the history of DDL and DML changes made to the table.

Prepare Upgrade Checklist

Clean Up: Recycle Bin

- Especially before patch set or release upgrade purge the recycle bin:
  - Since Oracle 12c this will be done by the preupgrade_fixups.sql

```
purge DBA_RECYCLEBIN;
```

- General recommendation:
  - Empty the recycle bin at least once per week with an automatic job during off-peak times

Components: Validation Check

- Make sure all components are VALID before upgrade
  - Components are INVALID?
  - If that does not correct component status, further diagnosis might be required

```
Select COMP_ID, COMP_NAME,
STATUS, VERSION from
DBA_REGISTRY where
STATUS<>'VALID';
```

```
@?/rdbms/admin/utlrp.sql
```

MOS Note:472937.1:
Information On Installed Database Components
MOS Note:753041.1:
How to diagnose Components with NON VALID status

Components: Removal

If you are not using it, think about removing it

Potential reasons to remove components:

- A component does not exist anymore in Oracle Database 12c

- Speed up the upgrade process

- A component is obsolete

[Compponent Removal Tech Details](http://tinyurl.com/ComponentCleanup)
http://tinyurl.com/ComponentCleanup

Have customers who save 90 days+

Performance: Preserve Statistics

Gather accurate performance statistics from production

- *Accurate* means: Starting at least 1 month before the upgrade
- *Use Automatic Workload Repository (AWR) Snapshots interval 30-60 minutes and retention >31 days*
  - *Extract AWR:*
  - *SQL> @?/rdbms/admin/awrextr.sql*

- Performance snapshot comparison using AWR DIFF reports:
  SQL>select * from table(
  DBMS_WORKLOAD_REPOSITORY.AWR_DIFF_REPORT_HTML(<DBID>, 1, 101, 121, <DBID>, 1, 201, 221));

MOS Note:1477599.1 Best Practices Around Data Collection For Performance Issues

Pre-Upgrade/Migrate Considerations

Always Consider Minimal or Zero Downtime Approach

- Set **Guaranteed Restore Points** accordingly
- Ensure **AWR retention** is long enough to capture historical data for comparison
  - ASH too
- Backup your database statistics, in case an older SQL plan needs restoring
- Definitely consider **RAT** for real workload testing, instead of synthetic workloads tests
- *If upgrading older version of RAC to new version in place, special consideration need to be understood*
  - *Upgrade the Grid Infrastructure stack first*
    - *Treat the GI upgrade like you do an OS upgrade*
  - Legacy clients may not support new Net connections

AWR - TopNSQL

https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_WORKLOAD_REPOSITORY.html#GUID-E2B46878-1BDB-4789-8A21-016A625530F1

Examples:

8 days retention, 60 minute interval, topnsql 30

- execute dbms_workload_repository.modify_snapshot_settings(retention => 11520, interval => 60, topnsql => 'DEFAULT');

8 days retention, 60 minute interval, topnsql 100

- execute dbms_workload_repository.modify_snapshot_settings(retention => 11520, interval => 60, topnsql => 100);

30 days retention, 60 minute interval, topnsql 100

- execute dbms_workload_repository.modify_snapshot_settings(retention => 43200, interval => 60, topnsql => 100);

90 days retention, 60 minute interval, topnsql 100

- execute dbms_workload_repository.modify_snapshot_settings(retention => 129600, interval => 60, topnsql => 100);

Verify AWR settings

- select a.snap_interval, a.retention, a.topnsql
- from dba_hist_wr_control a, v$database d
- where a.dbid = d.dbid;

Pre-Upgrade JAR Tips

- Set environment variables:

```
export NEW_HOME=$ORACLE_19C_HOME
export OLD_HOME=$ORACLE_HOME
```

- Run the pre-upgrade JAR:

```
$NEW_HOME/jdk/bin/java -jar $NEW_HOME/rdbms/admin/preupgrade.jar TERMINAL TEXT
```

- Perform fixups:

```
$NEW_HOME/perl/bin/perl -I$NEW_HOME/perl/lib \
   -I$NEW_HOME/rdbms/admin $NEW_HOME/rdbms/admin/catcon.pl \
   -l $ORACLE_BASE/cfgtoollogs/$ORACLE_SID/preupgrade/ \
   -b preup_$ORACLE_SID \
   $ORACLE_BASE/cfgtoollogs/$ORACLE_SID/preupgrade/preupgrade_fixups.sql
```

Pre-upgrade checks and review

- Force Logging
- Block Change Tracking
- NLS values:
  - Time zone
  - Language
  - Territory
  - Character set/ NLS character set
- Contents, size and space in FRA
- Restore Points
- Database links
- Directories
- External tables
- Jobs and Scheduler Jobs
- NOLOGGING objects and tablespaces
- Datafiles, Tempfiles, Redo Log files

- Environment variables
  - SQLPATH
  - TWO_TASK
- Check .profile, .bash_profile, .bashrc, etc.
- Check crontab
- Review deprecated, de-supported features
- OS links
- Upgrading to New Version of the OS (validate shell scripts and programs)
- Check Apex Compatibility
  - https://mikedietrichde.com/2017/05/02/is-your-apex-version-certified-with-your-database-release/

Sample PreUpgrade Check List

Pre Upgrade Tasks
Schedule the full database backups prior to upgrading / patching the environment (the night before the upgrade or patching)
Determine who will be available from the support side if we need IT support
Validate root, grid, oracle access, grid access
Export Database statistics for backup
Change AWR retention window to minimum **31 days** - need to do this way ahead of time
Review data guard configuration since all of the environments have data guard - Make sure that force logging is enabled - Check for corrupt datafiles from unrecoverable activities
Create a preemptive SR with Oracle
Stage the 19c software on PROD RAC
Stage the 19c software on DR RAC
Apply RU Patch PROD RAC to 19.10
Apply Patch DR RAC to 19.10
Setup SQL.NET ora with 11g logon
Copy initialization parameter file to 19c on PROD and DR
Copy password file to 19c on PROD and DR
Copy tnsnames.ora file
Execute dbupgdiag.sql
Execute Viscosity Pre Upgrade Check Script
Execute java with the preupgrade.jar file
Record invalid objects
Validate application server scripts for 19c database
Backup /etc/oratab
Review cron jobs
Make sure that glogin.sql is blank
Capture DBMS_JOBs
Capture directories
Capture extproc
Capture database links (as they need to be reset)
Day Before
Perform full level 0 backup of the Database (if possible)
Clean up file system space
Review output from preupgrade scripts and provide remediation plan

Pre-upgrade backup and capture

- Database configurations:
  - $ORACLE_HOME/dbs
  - $ORACLE_HOME/network/admin
  - Wallets
  - Diagnostic directories
- SGA and PGA TARGET ADVICE
- Opatch inventory, patch registry
- RMAN:
  - show all
  - report unrecoverable
  - report need backup
  - report schema
  - list backup
- Capture listener information
  - show log_directory
  - show log_file
  - show log_status
  - show trc_directory
  - show trc_file
  - show trc_level
  - show rawmode
  - show displaymode
  - show rules
  - show inbound_connect_timeout
  - show dynamic_registration
  - status services

Pre-upgrade tasks

- Upgrade or Remove APEX
  - If you are not using Apex, remove it
- Review and remove hidden parameters
- Remove OJVM if you are not using it
- Extend ADR retention policy (30-60 days ahead of the production upgrade cutover)
- Extend AWR retention policy (30-60 days ahead of the production upgrade cutover)

- Review contents of crontab
- Backup spfile and memory to pfile
- Backup /etc/oratab

- Generate IPCS reports

- Backup SAR directory

DBUA Tips

- Run DBUA in silent mode:

```
$NEW_HOME/bin/dbua -silent -dbName $ORACLE_SID \
                   -upgrade_parallelism 2 \
                   -recompile_invalid_objects true \
                   -upgradeTimezone true \
                   -performFixUp true
```

- Check upgrade status:

```
@$NEW_HOME/rdbms/admin/utlusts.sql TEXT
```

Auto Upgrade

Autoupgrade Tips

- Check Java version:

```
$ORACLE_HOME/jdk/bin/java -version
java version "1.6.0_75"
Java(TM) SE Runtime Environment (build 1.6.0_75-b13)
Java HotSpot(TM) 64-Bit Server VM (build 20.75-b01, mixed mode)
```

- Java must be 1.8 or greater (19c home Java can be used):

```
$NEW_HOME/jdk/bin/java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

Oracle Database 19c

My Oracle Support Document 2485457.1

19c ORACLE Database

- **Auto Upgrade** for Oracle Databases
  - Must download the AutoUpgrade Kit (Starting with 12.2 and 18.5)
    - For 12.2: Requires the January 2019 Release Updates (DBJAN2019RU)
  - Upgrade database instance from command line with **single configuration** file
  - Runs pre-upgrade tasks
  - Performs automated fix-ups
  - Performs the actual upgrade
  - Performs post-upgrade tasks
    - Automatic retry and tailback
    - Schedule the upgrade
    - Change init.ora parameters along the way

AutoUpgrade Tool
Doc ID 2485457.1

19c ORACLE Database

Source

- 11.2.0.4 or Higher
- The most recent version of AutoUpgrade Utility can be downloaded via this link: AutoUpgrade Tool 2485457.1.

Target

- Oracle Database 19c (19.3 and newer)
- Oracle Database 18c (18.5 and newer)
- Oracle Database 12c Release 2 (12.2 + DBJAN2019RU and newer)

Autoupgrade Tips

- Check Autoupgrade version:

```
$NEW_HOME/jdk/bin/java -jar $NEW_HOME/rdbms/admin/autoupgrade.jar -version
build.hash 04dd9f2
build.version 19.7.5
build.date 2020/02/11 15:28:49
build.max_target_version 19
build.type production
```

Autoupgrade Tips

- Sample autoupgrade configuration file:

```
# Global parameters
global.autoupg_log_dir=/opt/oracle/autoupgrade
# Database parameters
upg1.source_home=/opt/oracle/product/12.1.0.2/dbhome_1
upg1.target_home=/opt/oracle/product/19c/dbhome_1
upg1.sid=$ORACLE_SID
upg1.start_time=now
upg1.pdbs=*
upg1.log_dir=/opt/oracle/autoupgrade/$ORACLE_SID
upg1.upgrade_node=$(hostname -s)
upg1.run_utlrp=yes
upg1.timezone_upg=yes
upg1.target_version=19.7
```

Autoupgrade Tips

- Run in analysis mode:

```
$NEW_HOME/jdk/bin/java -jar $NEW_HOME/rdbms/admin/autoupgrade.jar \
                       -config /opt/oracle/autoupgrade/config.txt \
                       -mode analyze
```

- Run autoupgrade (deploy):

```
$NEW_HOME/jdk/bin/java -jar $NEW_HOME/rdbms/admin/autoupgrade.jar \
                       -config /opt/oracle/autoupgrade/config.txt \
                       -mode deploy
```

Autoupgrade Tips

- Sample autoupgrade configuration file:

```
# Global parameters
global.autoupg_log_dir=/opt/oracle/autoupgrade
# Database parameters
upg1.source_home=/opt/oracle/product/12.1.0.2/dbhome_1
upg1.target_home=/opt/oracle/product/19c/dbhome_1
upg1.sid=$ORACLE_SID
upg1.start_time=now
upg1.pdbs=*
upg1.log_dir=/opt/oracle/autoupgrade/$ORACLE_SID
upg1.upgrade_node=$(hostname -s)
upg1.run_utlrp=yes
upg1.timezone_upg=yes
upg1.target_version=19.7
```

Autoupgrade Tips

Another autoupgrade configuration file including PDB Migration

We have a lab for this!

```
cat << EOF > $ORADATA/autoupgrade/config.txt
# Global parameters
global.autoupg_log_dir=$ORADATA/autoupgrade
global.raise_compatible=yes
global.drop_grp_after_upgrade=yes
global.remove_underscore_parameters=yes

# Common database parameters
upg.upgrade_node=localhost
upg.source_home=$ORACLE_HOME
upg.sid=$ORACLE_SID
upg.start_time=now
upg.run_utlrp=yes
upg.timezone_upg=yes

EOF
```

```
if [ -d "$ORACLE_19C_HOME" ]
then cat << EOF >> $ORADATA/autoupgrade/config.txt
# Database parameters - 19c upgrade
upg.target_home=$ORACLE_19C_HOME
upg.target_version=19
EOF

elif [ -d "$ORACLE_21C_HOME" ]
then cat << EOF >> $ORADATA/autoupgrade/config.txt
# Database parameters - 21c upgrade
upg.target_home=$ORACLE_21C_HOME
upg.target_cdb=${ORACLE_SID}CDB
upg.target_pdb_name=${ORACLE_SID}PDB
upg.target_version=21.5
upg.target_pdb_copy_option=file_name_convert=NONE
EOF
```

APEX Upgrade

Apex Manual Upgrade

Oracle APEX **Release 22.2**: released on November 2022

18c Oracle Database

```
--   ORIGINAL PREUPGRADE ISSUE:
--   1.  Upgrade Oracle Application Express (APEX) manually before the database
--       upgrade.
--
--       The database contains APEX version 4.2.5.00.08. Upgrade APEX to at least
--       version 18.2.0.00.12.
--
--       Starting with Oracle Database Release 18, APEX is not upgraded
--       automatically as part of the database upgrade. Refer to My Oracle Support
--       Note 1088970.1 for information about APEX installation and upgrades.
```

```
SQL> select count(*) from APEX_040200.WWV_FLOWS where id = 4000;

  COUNT(*)
----------
```

📂 📄❾ development install → apexins.sql

- **October 22, 2020: apex_20.2.zip is out**
- **https://asktom.oracle.com/pls/apex/f?p=100:551::::RP,551:P551_CLASS_ID,P551_INVITED:9824,N&cs=110D22E6A6B683BC69BAFC2FE19677C37**

- Patch Set Bundle for Oracle APEX 20.2 (32006852)

https://www.oracle.com/tools/downloads/apex-v191-downloads.html

- apex_20.1.zip : Released April 2020
- apex_19.2_en.zip : November 2019

Upgrade Apex

@apexins.sql apex apex temp /i/

…. Thank you for installing Oracle Application Express 19.2.0.00.18

Oracle Application Express is installed in the APEX_190200 schema.

The structure of the link to the Application Express administration services is as follows:
http://host:port/pls/apex/apex_admin (Oracle HTTP Server with mod_plsql)
http://host:port/apex/apex_admin (Oracle XML DB HTTP listener with the embedded PL/SQL gateway)
http://host:port/apex/apex_admin (Oracle REST Data Services)

The structure of the link to the Application Express development interface is as follows:
http://host:port/pls/apex (Oracle HTTP Server with mod_plsql)
http://host:port/apex (Oracle XML DB HTTP listener with the embedded PL/SQL gateway)
http://host:port/apex (Oracle REST Data Services)

timing for: Phase 3 (Switch)
Elapsed: 00:00:52.44
timing for: Complete Installation
Elapsed: 00:10:57.93

PL/SQL procedure successfully completed.

1 row selected.

...null1.sql

timing for: Phase 1 (Installation)
Elapsed: 00:03:05.77
Phase 2 (Upgrade)

timing for: Enabling Phase 2
Elapsed: 00:00:00.00
#
 # Upgrade Metadata (1)
#
...reset_state_and_show_invalid.sql

timing for: Upgrade Metadata (1)
Elapsed: 00:01:42.03
#
Upgrade Metadata (2)
#
-- Upgrading new schema. -------

timing for: Upgrade Metadata (2)
Elapsed: 00:00:39.04
#
Recompiling APEX_190200 schema
#
...reset_state_and_show_invalid.sql

timing for: Recompiling APEX_190200 schema
Elapsed: 00:00:21.55
#
Configuring Restricted Schemas
#

Security

Blockchain Table
Immutable Table
Immutable Backups

Password
SQL Firewall

21c Started Out In Oracle Cloud

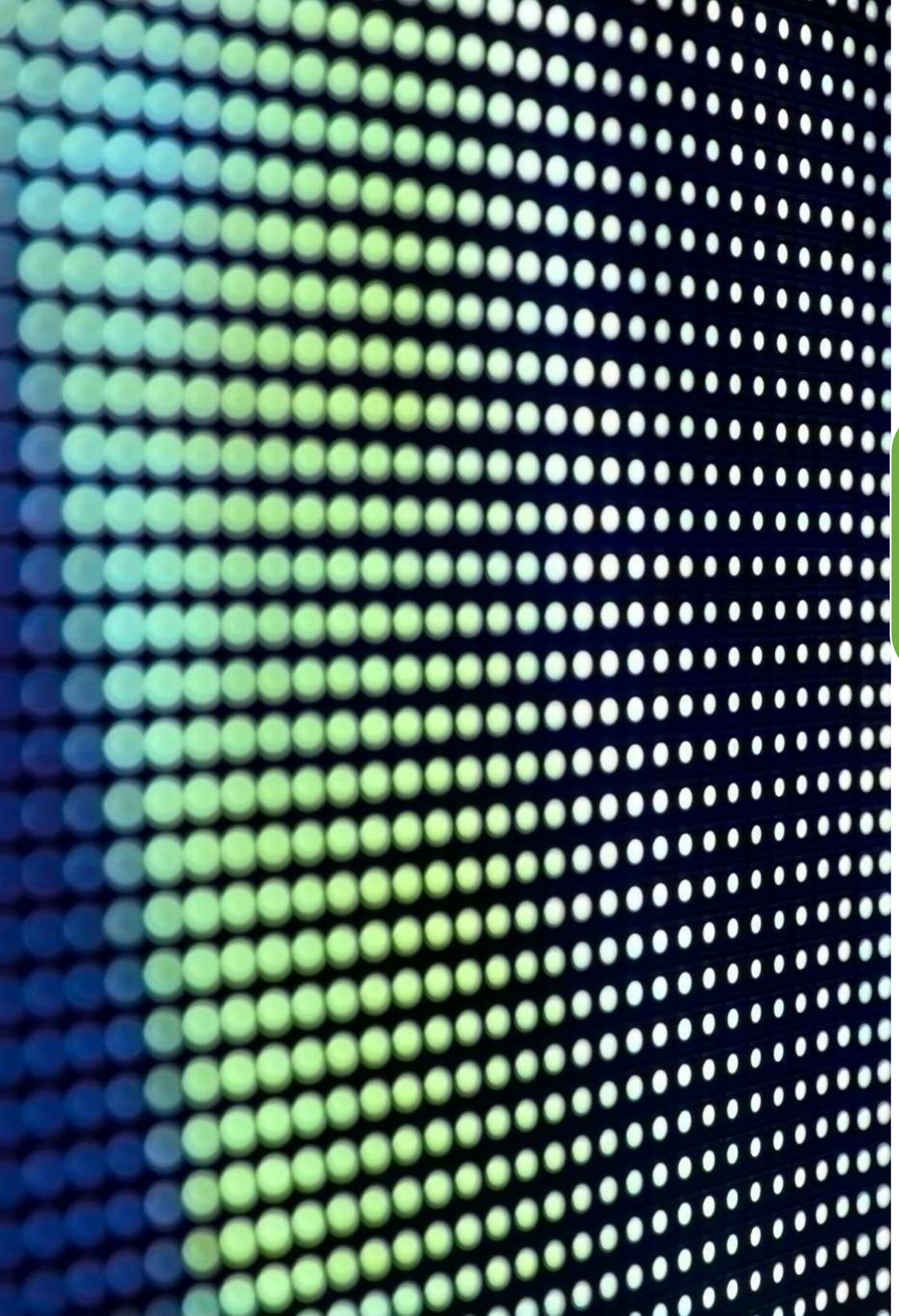
January 2021

Native blockchain table

- Secure ledger table managed by a trusted provider who has the best security in the world
- Create blockchain table [ledger_of_tables]
- Insert only
- Rows are cryptographically chained and trusted

```
CREATE BLOCKCHAIN TABLE account_ledger
(account VARCHAR2(100),
 account_date DATE,
 amount NUMBER)
        NO DROP UNTIL 60 DAYS IDLE
        NO DELETE LOCKED
        HASHING USING "SHA2_512" VERSION "v1";
```

Mandatory: `NO DROP`, `NO DELETE`, `HASHING USING`, and `VERSION`



Blockchain Table in 21c

Backported to 19c

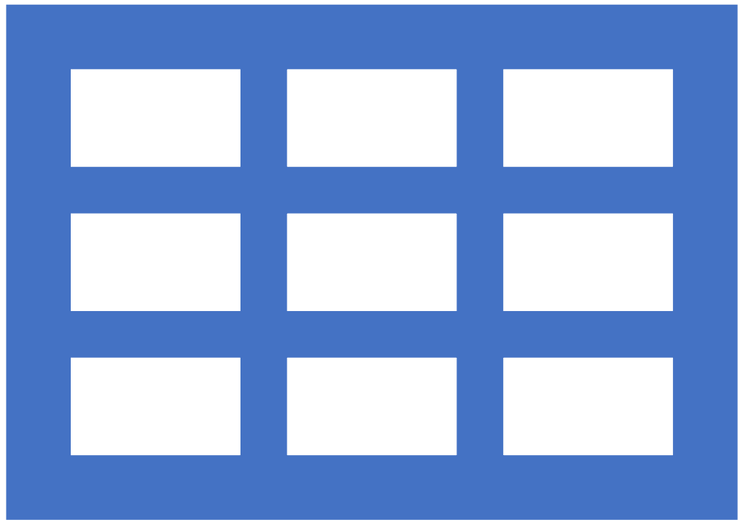
Requires 19.11 compatible

A blockchain table is:

- an insert-only table with an associated table-level and row-level retention period
- tamper proof

Rows are organized into chains

- each row contains the hash of the data contained in the row plus the hash of the previous rows data



Immutable tables

Immutable tables were introduced to Oracle 21.3 and 19.11 at the same time, so it could be considered a 19c and 21c new feature.

Immutable tables are insert-only tables in which existing data cannot be modified

Comparison

Immutable Tables Versus Blockchain Tables

Immutable Tables	Blockchain Tables
Prevents unauthorized changes by rogue or compromised insiders who have access to user credentials.	Prevents unauthorized changes by rogue or compromised insiders • Detects unauthorized changes made by bypassing Oracle Database software • Detects end user impersonation and insertion of data in a user's name but without their authorization • Prevents data tampering and ensures that data was actually inserted in to the table
Rows are not chained together.	Each row, except the first row, is **chained to the previous row** by using a **cryptographic hash**. The hash value of a row is computed based on the row data and the hash value of the previous row in the chain. Any modification to a row breaks the chain, thereby indicating that the row was tampered.
Inserting rows does not require additional processing at commit time.	Additional processing time is required, at commit time, to chain rows.

Only rows that are outside the specified retention period can be deleted from an immutable table.

The SYS user or the owner of the schema can delete immutable table rows.

Use the DBMS_IMMUTABLE_TABLE.DELETE_EXPIRED_ROWS procedure to delete all rows that are beyond the specified retention period or obsolete rows that were created before a specified time.

CREATE IMMUTABLE TABLE LOANS(id NUMBER, accountNo NUMBER, luser VARCHAR2(40), value NUMBER) **NO DROP UNTIL 40 DAYS** IDLE **NO DELETE UNTIL 100 DAYS** AFTER INSERT;

No drop and no delete are mandatory options

23c: Blockchain

- Blockchain Table **User Chains**
  - Earlier versions of blockchain tables supported only **system** chains.
  - Now we can have a user chain: chain of rows based on a set of up to three user-defined columns
  - Increases the **flexibility** of applying blockchain tables and their **verification procedures**
- Blockchain Table Row Versions
  - See multiple historical versions of a row
- **Blockchain Table Log History** with **Flashback Data Archive History**
  - Each change in a regular table will be added to the blockchain log history table as a separate row within a cryptographic hash chain
- Add and Drop User Columns in Blockchain and Immutable Tables
- Blockchain Table Countersignature (at the time of signing a row)
- Blockchain Table Delegate Signer
  - A delegate is an alternate user who's allowed to sign rows inserted by the primary user. This feature allows a delegate to sign rows in an immutable or blockchain table on behalf of another user.
- New Special Privilege Required to **Set Long Idle Retention Times** for Blockchain and Immutable Tables
  - To fight disk space exhaustion attack

RMAN Immutable Backups to OCI

23c: Security Passwords

- Increased Oracle Database Password Length
  - With 23c, Oracle supports passwords up to **1024 bytes** in length.
  - For previous releases, the password length and the secure role password length could be up to 30 bytes.

- Oracle Data Pump Export and Import Support for Longer Database User Passwords
  - Oracle Data Pump can export and import database users that have passwords up to **500 characters** long.

23c: Changes in Encryption Algorithms and Default Modes

Encryption algorithm changes:

- The default column encryption for both AES and ARIA is now 256.
  - The previous default for TDE column encryption was AES192.
  - For TDE tablespace encryption, the default was AES128.
- The decryption libraries for the GOST and SEED algorithms are deprecated.

- Recovery Manager (RMAN) integrity check for column encryption keys now uses SHA512 instead of SHA1.
  - The keys for Oracle RMAN and column keys are now derived from SHA512/AES for key generation.
  - Previous releases used SHA-1/3DES as a pseudo-random function.

- **Transport Layer Security (TLS) 1.3 Protocol** Now Supported in Oracle Database

Quarantine for Execution Plans for SQL Statements

19c ORACLE Database

ORACLE EXADATA

- **SQL statements terminated** due to excessive CPU and I/O **can be quarantined**
- **Execution Plans** for above SQL **are Quarantined**

- Enable/Disable a quarantine configuration:
- DBMS_SQLQ.ALTER_QUARANTINE procedure

- Drop & Alter Quarantined Configuration (*unused quarantined configs deleted 53 weeks*):
- DBMS_SQLQ.**DROP**_QUARANTINE & DBMS_SQLQ.**ALTER**_QUARANTINE

- **DBA_SQL_QUARANTINE** - details of quarantine configurations
- V$SQL & GV$SQL to get execution plan details of quarantined SQL

18c Oracle Database

18c: Manually kill a statement without breaking the session:
•ALTER SYSTEM CANCEL SQL 'SID, SERIAL, @INST_ID, SQL_ID';

23c: SQL Firewall

- B**uild an allow-list of SQL statements** that a typical application user performs, and then detects, blocks, and logs any unexpected SQL.
- Detect anomalies and prevent SQL injection attacks
- SQL Firewall examines all SQL, including **session context information such as IP address and OS user** to enforce protection
- SQL Firewall logs and (if enabled) blocks unauthorized SQL.
  - SQL Firewall is embedded into the database kernel, ensuring that it cannot be bypassed.

- By enforcing an **allow-list of SQL** and approved session contexts, SQL Firewall can prevent many zero-day attacks.
- You can use SQL Firewall in both the root and a pluggable database (PDB).
- Query the **DBA_SQL_FIREWALL_VIOLATIONS** data dictionary view for violations that SQL Firewall enforced for the user to find any abnormal access pattern or unauthorized SQL.
- Can export and import SQL Firewall metadata, including existing allow-lists, by using the Oracle Data Pump

23c: SQL Firewall Packages and Views

- Enable SQL Firewall: EXEC **DBMS_SQL_FIREWALL.**ENABLE;
  - DBMS_SQL_FIREWALL.CREATE_CAPTURE
  - DBMS_SQL_FIREWALL.START_CAPTURE
  - DBMS_SQL_FIREWALL.STOP_CAPTURE
  - DBMS_SQL_FIREWALL.GENERATE_ALLOW_LIST
  - DBMS_SQL_FIREWALL.ENABLE_ALLOW_LIST
  - DBMS_SQL_FIREWALL.ADD_ALLOWED_CONTEXT
- **DBA_SQL_FIREWALL_*** data dictionary views:
  - DBA_SQL_FIREWALL_VIOLATIONS
  - DBA_SQL_FIREWALL_ALLOWED_SQL
  - DBA_SQL_FIREWALL_ALLOWED_OS_PROG
  - DBA_SQL_FIREWALL_CAPTURE_LOGS

Follow Us Online!


Facebook.com/ViscosityNA


Linkedin.com/company/Viscosity-North-America


@ViscosityNA

Viscosity North America


Facebook.com/ViscosityNA


@Viscosity_NA