Managing UNDO Data

EMP table

empid	sal
100	5000

UPDATE EMP SET SAL=6000 where empid=100;

- The Oracle Database server saves the old value (undo data) when a process changes data in a database.
- It stores the data as it exists before modifications.
- Retained at least until the transaction is ended. (next slide)

UNDO Data used to support :

- Rollback operations
- Read-consistent queries
- Oracle Flashback Query, Oracle Flashback Transaction, and Oracle Flashback Table
- Recovery from failed transactions

A failed transaction occurs when a user session ends abnormally (possibly because of network errors or a failure on the client computer) before the user decides to commitor roll back the transaction.
Failed transactions may also occur when the instance crashes or you issue the `SHUTDOWN ABORT` command.

Managing UNDO Data

How the transaction ends?

- User undoes a transaction (transaction rolls back).
- User ends a transaction (transaction commit).
- User executes a DDL statement, such as a CREATE, DROP, RENAME, or ALTER statement.
- session terminates abnormally (transaction rolls back).
- User session terminates normally with an exit (transaction commits).

Managing UNDO Data

Transactions and Undo Data

- Each transaction is assigned to only one undo segment.
- An undo segment can service more than one transaction at a time.
- When a transaction starts, it is assigned to an undo segment. Throughout the life of the transaction, when data is changed, the original (before the change) values are copied into the undo segment. You can see which transactions are assigned to which undo segments by checking the V$TRANSACTION dynamic performance view.
- Undo segments are specialized segments that are automatically created by the database server as needed to support transactions.
  Like all segments, undo segments are made up of extents, which, in turn, consist of data blocks. Undo segments automatically grow and shrink as needed, acting as a circular storage buffer for their assigned transactions.
- Undo information is stored in undo segments, which are stored in an undo tablespace.
- you cannot create other segment types, such as tables, in the undo tablespace

Managing UNDO Data

Transactions and Undo Data (self-study)

The Database Configuration Assistant (DBCA) automatically creates a smallfile undo tablespace. You can also create a bigfile undo tablespace. However, in a high-volume online transaction processing (OLTP) environment with many
short concurrent transactions, contention could occur on the file header. An undo tablespace, stored in multiple datafiles, can resolve this potential issue.
Although a database may have many undo tablespaces, only one of them at a time can be designated as the current undo tablespace for any instance in the database.

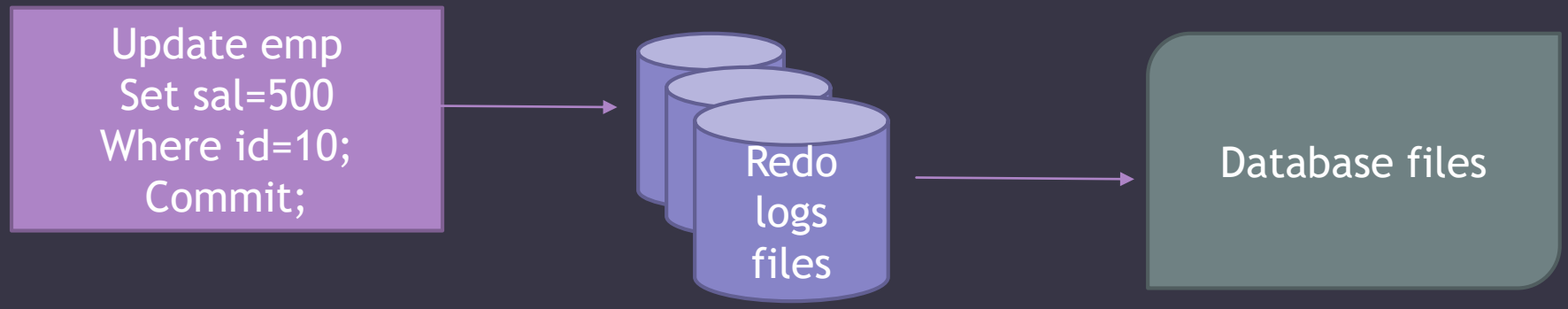
Undo segments are automatically created and always owned by `SYS`. Because the undo segments act as a circular buffer, each segment has a minimum of two extents. The default maximum number of extents depends on the database block size but is very high (32,765 for an 8 KB block size).
Undo tablespaces are permanent, locally managed tablespaces with automatic extent allocation. They are automatically managed by the database.
Because undo data is required to recover from failed transactions (such as those that may occur when an instance crashes), undo tablespaces can be recovered only while the instance is in the MOUNT state.

Managing UNDO Data

Comparing Undo Data and Redo Data

Redo log files

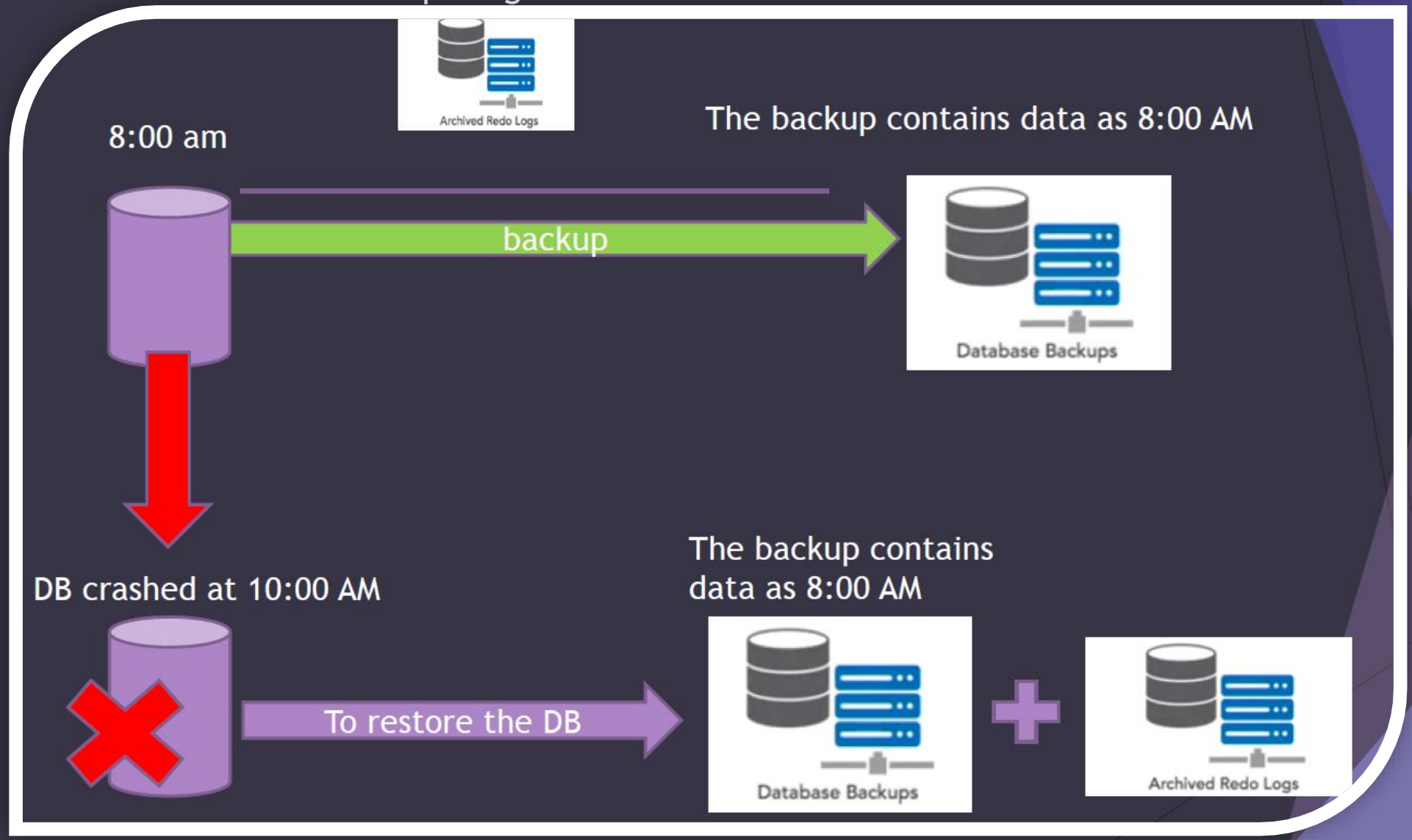
stores changes to the database as they occur and are used for data recovery.



	Undo	Redo
Used for	Rollback, read consistency, flashback	Rolling forward of database changes
Stored in	Undo segments	Redo Log files

Managing UNDO Data

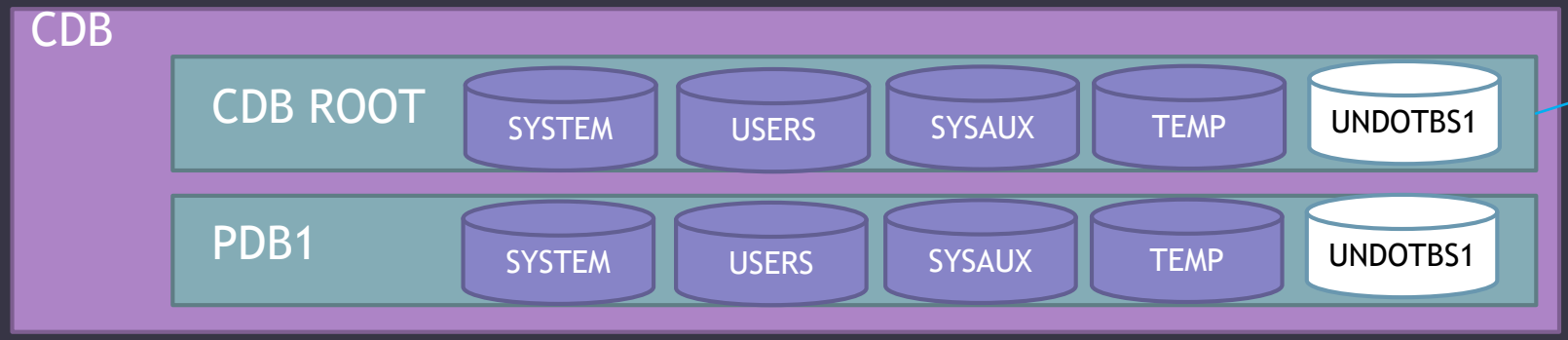
Comparing Undo Data and Redo Data



Managing UNDO Data

Local Undo Mode Versus Shared Undo Mode

Two UNDO modes: SHARED versus LOCAL



LOCAL_UNDO_ENABLED= true
This mean each pluggable can have its own undo

- There is only one shared UNDO tablespace (in CDB root).
- There can be a local UNDO tablespace in each PDB
- You can set a CDB in local UNDO mode either at CDB creation or by altering the CDB property.

table :DATABASE_PROPERTIES property LOCAL_UNDO_ENABLED= true note: it is true by default in release 18c

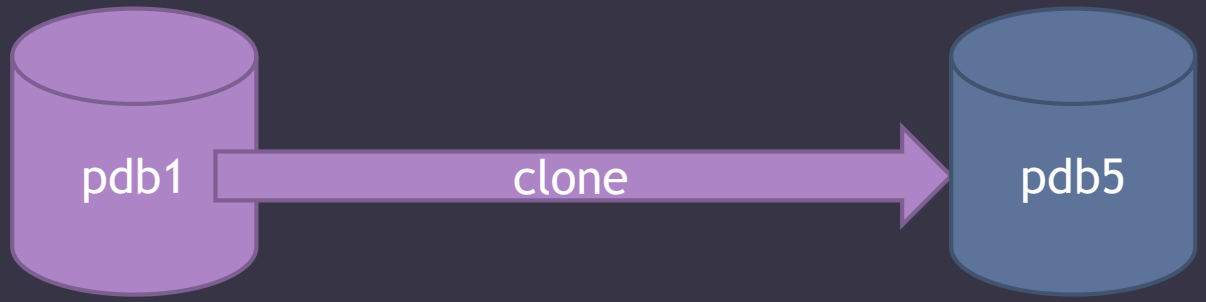
When is local UNDO mode required?

- Hot cloning
- Near-zero downtime PDB relocation

```
SQL> STARTUP UPGRADE;
SQL> ALTER DATABASE LOCAL UNDO ON;
```

Managing UNDO Data

When the property LOCAL_UNDO_ENABLED= true



- No errors in the clone
- PDB5 will be same PDB1
- uncommitted trans will not be cloned

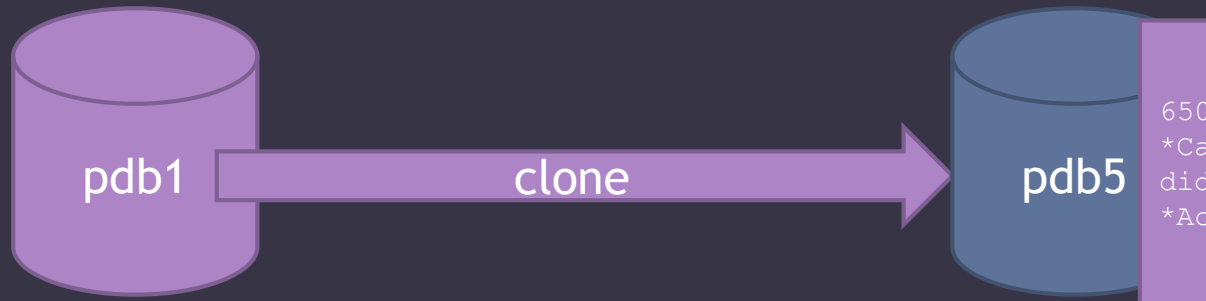
contains some uncommitted trans.

For details back to lessons

Cloning PDBs part 1
Cloning PDBs part 2

When the property LOCAL_UNDO_ENABLED= false

Clone will fail and give error ORA-65035




```
65035, 00000, "unable to create pluggable database from %s"
*Cause: An attempt was made to clone a pluggable database that
did not have local undo enabled.
*Action: Enable local undo for the PDB and retry the operation.
```

contains some uncommitted trans.

Managing UNDO Data

Overview of Automatic Undo Management

- Oracle provides a fully automated mechanism for managing undo information and space.
- the database manages undo segments in an undo tablespace.
- Automatic undo management is the default mode for a newly installed database. An auto-extending undo tablespace named UNDOTBS1 is automatically created when you create the database with Database Configuration Assistant (DBCA).
- When the database instance starts, the database automatically selects the first available undo tablespace. If no undo tablespace is available, then the instance starts without an undo tablespace and stores undo records in the SYSTEM tablespace. This is not recommended, and an alert message is written to the alert log file to warn that the system is running without an undo tablespace.

Initialization Parameter	Description
UNDO_MANAGEMENT	If AUTO or null, enables automatic undo management. If MANUAL, sets manual undo management mode. The default is AUTO.
UNDO_TABLESPACE	Optional, and valid only in automatic undo management mode. Specifies the name of an undo tablespace. Use only when the database has multiple undo tablespaces and you want to direct the database instance to use a particular undo tablespace.

Space management for rollback segments is complex. Oracle strongly recommends leaving the database in automatic undo management mode.

Managing UNDO Data

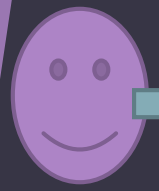
About the Undo Retention Period

When automatic undo management is enabled, there is always a current **undo retention period**, which is the minimum amount of time that Oracle Database attempts to retain old undo information before overwriting it.

After a transaction is committed, undo data is no longer needed for rollback or transaction recovery purposes. However, for consistent read purposes, long-running queries may require this old undo information for producing older images of data blocks. Furthermore, the success of several Oracle Flashback features can also depend upon the availability of older undo information. For these reasons, it is desirable to retain the old undo information for as long as possible.

UNDO_RETENTION
The default is 900s
Which is 15m

User A



Select emp_id,sal from emp

Assume that this query will take 5min
So it will finish at 10:05

10:00 AM

The user A should see the data as 10:00 exactly
And this is called read consistency

User B

Update emp Set sal=0;

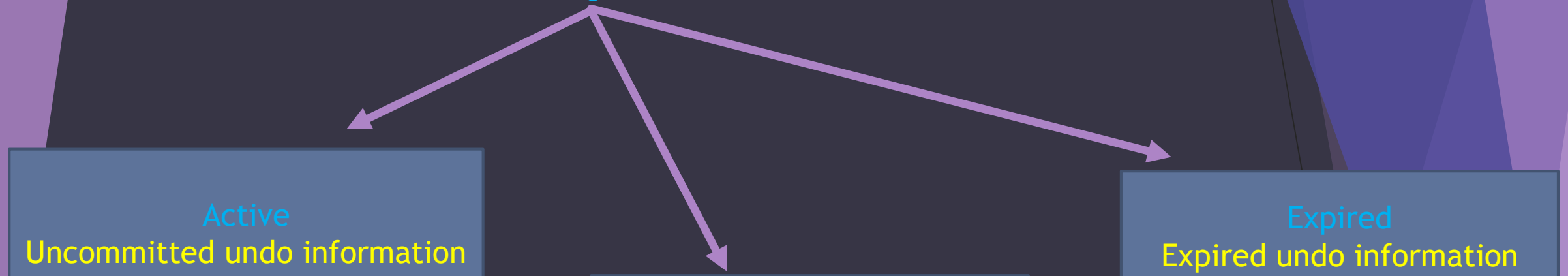
It will take less than 1s

10:03 AM

This will not reflect the user A query
User A still see the old data, because he execute the query at 10:00 am

Managing UNDO Data

Categories of Undo



Active

Uncommitted undo information

required if a user wants to roll back or if the transaction has failed.

Uncommitted undo information is never overwritten.

Unexpired

Committed undo information

Old undo information with an age that is less than the current undo retention period

retained for consistent read and Oracle Flashback operations

Expired

Expired undo information

Old (committed) undo information that is older than the current undo retention period is said to be *expired* and its space is available to be overwritten by new transactions.

Managing UNDO Data

Automatic Tuning of Undo Retention

Oracle Database automatically tunes the undo retention period based on how the undo tablespace is configured.

- If the undo tablespace is configured with the `AUTOEXTEND` option, the database dynamically tunes the undo retention period to be somewhat longer than the longest-running active query on the system. However, this retention period may be insufficient to accommodate Oracle Flashback operations. Oracle Flashback operations resulting in `snapshot too old` errors are the indicator that you must intervene to ensure that sufficient undo data is retained to support these operations. To better accommodate Oracle Flashback features, you can either set the `UNDO_RETENTION` parameter to a value equal to the longest expected Oracle Flashback operation, or you can change the undo tablespace to fixed size.
- If the undo tablespace is fixed size, the database dynamically tunes the undo retention period for the best possible retention for that tablespace size and the current system load. This best possible retention time is typically significantly greater than the duration of the longest-running active query.

  If you decide to change the undo tablespace to fixed-size, you must choose a tablespace size that is sufficiently large. If you choose an undo tablespace size that is too small, the following two errors could occur:

  - DML could fail because there is not enough space to accommodate undo for new transactions.
  - Long-running queries could fail with a `snapshot too old` error, which means that there was insufficient undo data for read consistency.

It ignore the value of undo_retention Unless you enable retention guarantee.

Self study

Retention Guarantee

To guarantee the success of long-running queries or Oracle Flashback operations, you can enable retention guarantee.

If retention guarantee is enabled, then the specified minimum undo retention is guaranteed; the database never overwrites unexpired undo data even if it means that transactions fail due to lack of space in the undo tablespace. If retention guarantee is not enabled, then the database can overwrite unexpired undo when space is low, thus lowering the undo retention for the system. This option is disabled by default.

> **WARNING:**
>
> Enabling retention guarantee can cause multiple DML operations to fail. Use with caution.

You enable retention guarantee by specifying the `RETENTION GUARANTEE` clause for the undo tablespace when you create it with either the `CREATE DATABASE` or `CREATE UNDO TABLESPACE` statement. Or, you can later specify this clause in an `ALTER TABLESPACE` statement. You disable retention guarantee with the `RETENTION NOGUARANTEE` clause.

You can use the `DBA_TABLESPACES` view to determine the retention guarantee setting for the undo tablespace. A column named `RETENTION` contains a value of `GUARANTEE`, `NOGUARANTEE`, or `NOT APPLY`, where `NOT APPLY` is used for tablespaces other than the undo tablespace.