➢ Undo Table spaces

1. Introduction to Automatic Undo Management
2. UNDO_RETENTION
3. Managing Undo Tablespaces

➢ Temporary Tablespaces

1. Default Temporary Tablespace
2. Restrictions on Default temporary tablespace

DEMO –

- ➢ Practical on Undo Table space
- ➢ Practical on Temp Table space

1. Undo Tablespace:

Every Oracle Database must have a method of maintaining information that is used to roll back, or undo, changes to the database. Such information consists of records of the actions of transactions, primarily before they are committed. These records are collectively referred to as undo.

Undo records are used to:

- Roll back transactions when a ROLLBACK statement is issued
- Recover the database
- Provide read consistency
- Analyze data as of an earlier point in time by using Oracle Flashback Query
- Recover from logical corruptions using Oracle Flashback features

1.1. Introduction to Automatic Undo Management

Oracle provides a fully automated mechanism, referred to as automatic undo management, for managing undo information and space. In this management mode, you create an undo tablespace, and the server automatically manages undo segments and space among the various active sessions.

SET UNDO_MANAGEMENT parameter to AUTO to enable automatic undo management.

1.2. UNDO_RETENTION

The UNDO_RETENTION parameter specifies the amount of redo information to be saved in the undo tablespace before it can be overwritten. The appropriate value for this parameter depends on the size of the undo tablespace and the nature of the queries in your database. If the queries aren'thuge, they don't need to have large snapshots of data, and you could get by with a low undo retentioninterval. Similarly, if there is plenty of free space available in the undo tablespace, transactions won't be overwritten, and you are less likely to run into the snapshot-too-old problem.

If you plan on using the Flashback Query feature extensively, you will have to figure out how far back in time your Flashback queries will go, and specify the UNDO_RETENTION parameter accordingly.

Default value: 900 (seconds)

Parameter type: Dynamic.

You can use the ALTER SYSTEM command to increase the value to a practically unlimited time period.

1.3. Undo Tablespaces

- Creating an Undo Tablespace
- Altering an Undo Tablespace
- Dropping an Undo Tablespace
- Switching Undo Tablespaces
- Establishing User Quotas for Undo Space
- Undo Space Data Dictionary Views

1.3.1. Creating an Undo Tablespace

We can create undo tablespace in two process

One is using CREATE DATABASE while we go for UNDO_MANGAMENT=AUTO option. This is used while we are creating a new database.

Other is using CREATE UNDO TABLESPACE statement.

We cannot create database objects in undo tablespace. It is reserved for system managed undo data.

1.3.2. Altering an UNDO Tablespaces

Undo tablespaces can be altered using the statement ALTER TABLESPACE.

We alter tablespaces as

- Adding a datafile
- Renaming a datafile
- Brining a datafile online or taking offline
- Beginning or ending an open backup on a datafile
- Enabling and Disabling undo retention guarantee

For example we want add a datafile, when assigned tablespace ran out of space.

Alter tablespace <tablespace name>add datafile <location of datafile> autoextend <on/off> next <size> maxsize <options>

1.3.3. Dropping an Undo Tablespace

We can drop the tablespace using DROP TABLESPACE statement.

Drop tablespace <tablespace name>

We can drop undo tablespace only when it is not in use.

1.3.4. Switching an Undo Tablespace

We can switch from one undo tablespace to other. By using ALTER SYSTEM SET we can change the undo tablespace.

While changing if error occurs then it is caused due to

1. If tablespace does not exists
2. If the given tablespace is not undo tablespace
3. If the tablespace is used by other instance (In RAC)

Syntax: alter system set undo_tablespace = <tablespace name>;

Since it is dynamic parameter we can change when database is open.

The database is online while the switch operation is performed, and user transactions can be executed while this command is being executed. When the switch operation completes successfully, all transactions started after the switch operation began are assigned to transaction tables in the new undo tablespace. The switch operation does not wait for transactions in the old undo tablespace to commit. If there are any pending transactions in the old undo tablespace, the old undo tablespace enters into a PENDING OFFLINE mode (status). In this mode, existing transactions can continue to execute, but undo records for new user transactions cannot be stored in this undo tablespace.

1.4. Views for UNDO TABLESPACE

View	Description
V$UNDOSTAT	Contains statistics for monitoring and tuning undo space. Use this view to help estimate the amount of undo space required for the current workload. The database also uses this information to help tune undo usage in the system. This view is meaningful only in automatic undo management mode.
V$ROLLSTAT	For automatic undo management mode, information reflects behavior of the undo segments in the undo tablespace
V$TRANSACTION	Contains Undo segment information
DBA_UNDO_EXTENTS	Shows the status and size of each extent in the undo tablespace.
DBA_HIST_UNDOSTAT	Contains statistical snapshots of v$undostat information.

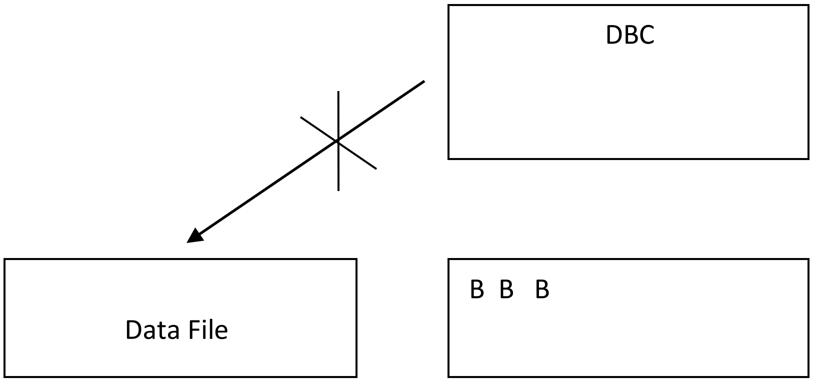
Note: The undo blocks occupied by a transaction will be expired once the transaction commits.

Error frequently gets in undo tablespaces **is ORA-1555 error**

This error occurs for following scenario

Let us assume three conditions

1. T1 → Updating table A and commit
2. T2 → updating table B
3. T3 → Selecting data from A



- In the above condition T1 make update on table A and committed. Because of this dirty blocks are generated in DBC and undo blocks are used from undo tablespace.
- T2 is updating table B and because of non-availability of undo blocks T2 override expired undo blocks of T1
- T3 is selecting the data from A. This operation will first look for data in undo tablespace, but already blocks of A are occupied by B performed by T2, it will not retrieve any data. Then it will check for latest data in datafiles, but as dirty blocks are not yet written to datafiles, there are transaction will be unable to get data. In this situation it throws the error ORA-1555

Imp points to remember:

The undo blocks occupied by a transaction will become expired once the transaction commits.
The data will be selected from undo tablespace, if any DML operation is being performed on the table on which select query is also fired. This is to maintain read consistency.

Solutions

1. Re-issue the SELECT statement will be a solution.
2. It may occur due to undersized undo tablespace. So increase undo tablespace.
3. Increasing undo_retention value
4. Avoiding frequent commits

Note: Don't ever allow undo & Temp tablespaces to be in AUTOEXTEND ON

2. Temporary Tablespaces

A **temporary tablespace** contains temporary data that perseveres only for the duration ofthe session. Temporary tablespaces can improve the concurrency of multiple sort operations that do not fit in memory and can improve the efficiency of space management operations during sorts.

Temporary tablespaces are used to store the following:

- Intermediate sort results
- Temporary tables and temporary indexes
- Temporary LOBs
- Temporary B-trees

Temporary Tablespace cannot contain any permanent objects.

Create temporary tablespace <tablespace name> tempfile <temp file location> size <bytes> extent management <local/auto> uniform size <bytes>;

Locally managed temporary tablespaces have temporary data files (temp files), which are similar to ordinary data files except that:

- Tempfiles are always set to the NOLOGGING mode.
- You cannot make a tempfile read-only.
- You cannot rename a temp file.
- You cannot create a tempfile with the ALTER DATABASE command.
- Temp files are required for read-only databases.
- Media recovery does not recover tempfiles.
- To optimize the performance of a sort in a temporary tablespace, set the UNIFORM SIZE to multiple of the parameter SORT_AREA_SIZE.

2.1. Default Temporary Tablespace

Specifies a database to use temporary tablespace by eliminating system tablespaces for storing temporary data.

Default temporary tablespace can be done by using
Create database
Alter database

- After creating a database, a default temporary tablespace can be set by following commands
- Alter database default temporary tablespace <temporary tablespace name>;
- To find the default temporary tablespace from the database query from the table database_properties

2.2. Restrictions on Default temporary tablespace

Dropping a default temporary tablespace

We cannot drop a default temporary tablespace until after a new default temporary tablespace is created. Users assigned to default temporary tablespace will automatically moves to the new default temporary tablespace.

Changing the Type of a Default Temporary Tablespace

Because a default temporary tablespace must be either the SYSTEM tablespace or a temporarytablespace, you cannot change the default temporary tablespace to a permanent type.

Taking Default Temporary Tablespace Offline

Tablespaces are taken offline to make that part of the database unavailable to other users, for an offline backup, maintenance or making any changes to applications will not affect the temporary tablespace, so we cannot take a default temporary tablespace as offline.

DEMO

1. UNDO TABLESPACE

1.1. Creating Undo TBS

```
Sql>CREATE UNDO TABLESPACE undotbs_02 DATAFILE '/u01/oracle/rbdb1/undo0201.dbf'      SIZE 20M
AUTOEXTEND ON;

Sql>ALTER TABLESPACE undotbs_02 ADD DATAFILE '/u01/user18/test/undo0102.dbf' size 10m
AUTOEXTEND ON;
```

1.2. Switching Undo Tablespaces

```
Sql>ALTER SYSTEM SET UNDO_TABLESPACE = undotbs_02   SCOPE=both;
```

1.3. Views for undo tablespace

- V$undostat
- V$rollstat
- V$transaction
- dba_undo_extents

2. View for Sysaux tablespace:

- V$sysaux_occupants

3. TEMPORARY TABLESPACES

3.1. Creating temporary tablespace

```
Sql>CREATE TEMPORARY TABLESPACE hztemp TEMPFILE ‘/u02/oracle/data/hztemp01.dbf' SIZE 20m ;
```

3.2. Changing default temporary tablespace of database

```
Sql>ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp1;
```

- Here default temporary tablespace is changed to temp1

3.3. Creating temporary tablespace group

```
Sql>CREATE TEMPORARY TABLESPACE temp1 TEMPFILE  '/u02/user18/test/data/temp01.dbf' size 50m
TABLESPACE GROUP tmpgrp;
Sql>select * from  dba_tablespace_groups;
```

3.4. Adding / Removing a tablespace from a group :

```
Sql> ALTER TABLESPACE hztemp TABLESPACE GROUP TMPGRP;
Sql>ALTER TABLESPACE hztemp  TABLESPACE GROUP  '';
```

3.5. Changing member of a tablespace group

```
Sql>ALTER TABLESPACE hztemp2 TABLESPACE GROUP group2;
```

3.6. Assigning a Tablespace Group as the Default Temporary Tablespace

```
Sql>ALTER DATABASE DEFAULT TEMPORARY TABLESPACE group2;
```

3.7. Creating bigfile tablespace

```
Sql>CREATE BIGFILE TABLESPACE bigtbs DATAFILE ‘/u02/user18/test/data/bigtbs01.dbf’ SIZE   3g;
```

3.8. Views for temporary tablespace

- v$tempfile
- dba_temp_files
- dba_tablespace_groups