Managing Data Recovery

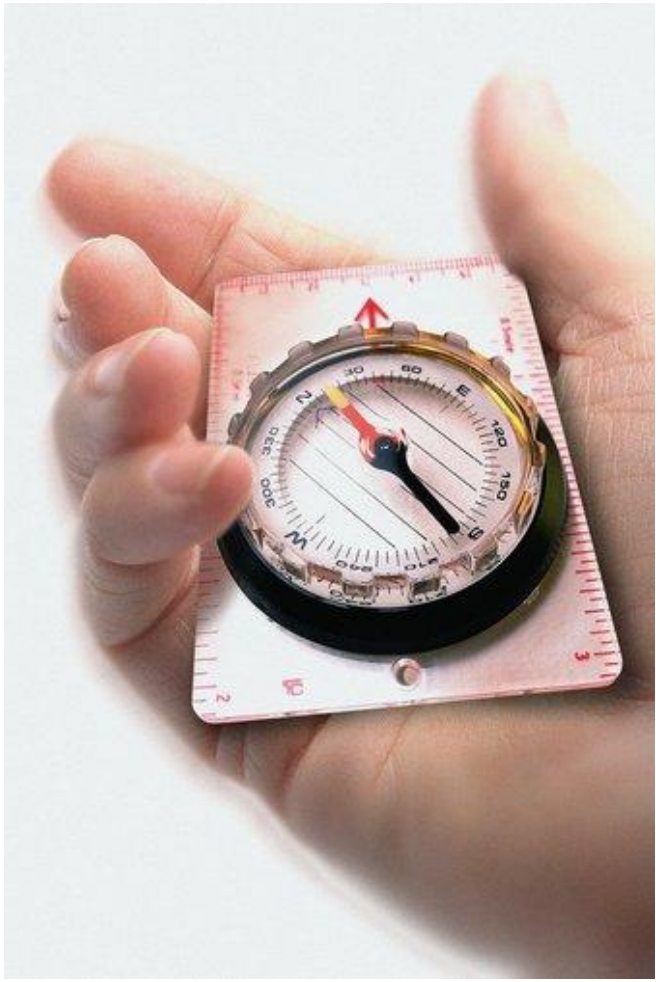
THE INTERNATIONAL INSTITUTE OF
SUPINFO
INFORMATION TECHNOLOGY

www.supinfo.com

SUPINFO OFFICIAL
SOC
CURRICULUM

Course objectives

By completing this course, you will be able to:



- Realize different recovery
- Explain reasons for incomplete recovery
- Monitor the Flashback Database
- Perform transaction level recovery using Flashback Transaction query

Course topics

Course's plan:

- Recovering from Noncritical Losses
- Database Recovery
- Flashback Database
- Recovering from User Errors

Recovering from Noncritical Losses

Preview

- Recover temporary tablespaces
- Recover a redo log group member
- Recover index tablespaces
- Recover read-only tablespaces
- Re-create the password file

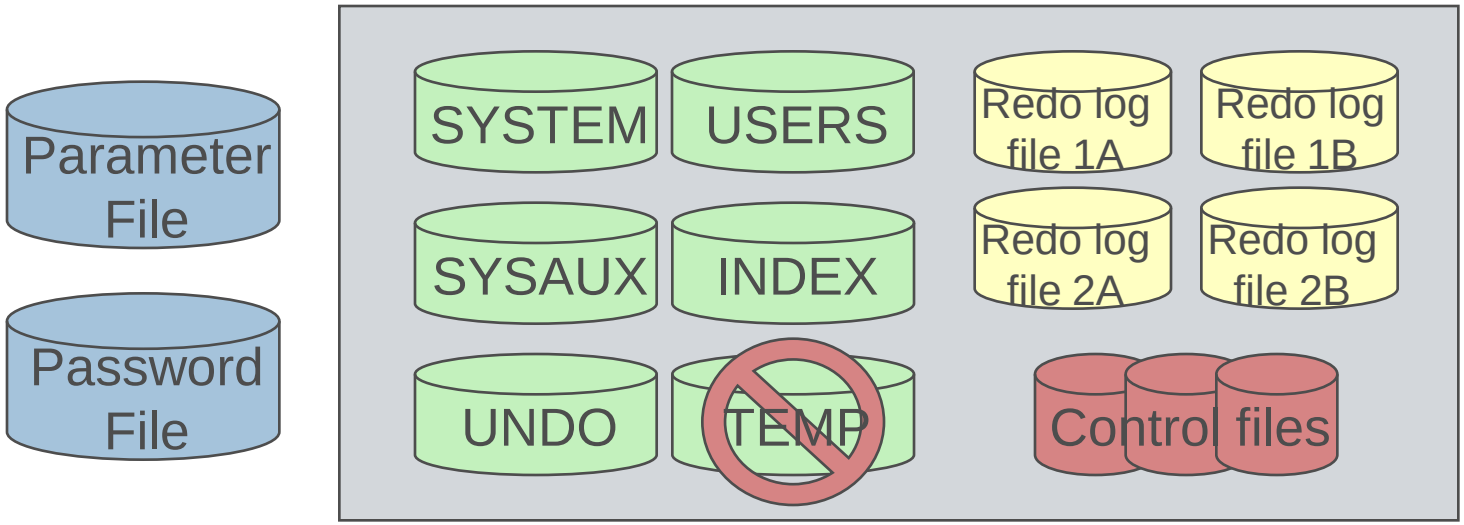
Recover temporary tablespaces

Recovery of Noncritical Files

- Create a new file
- Rebuild the file
- Recover the lost or damaged files

Recover temporary tablespaces

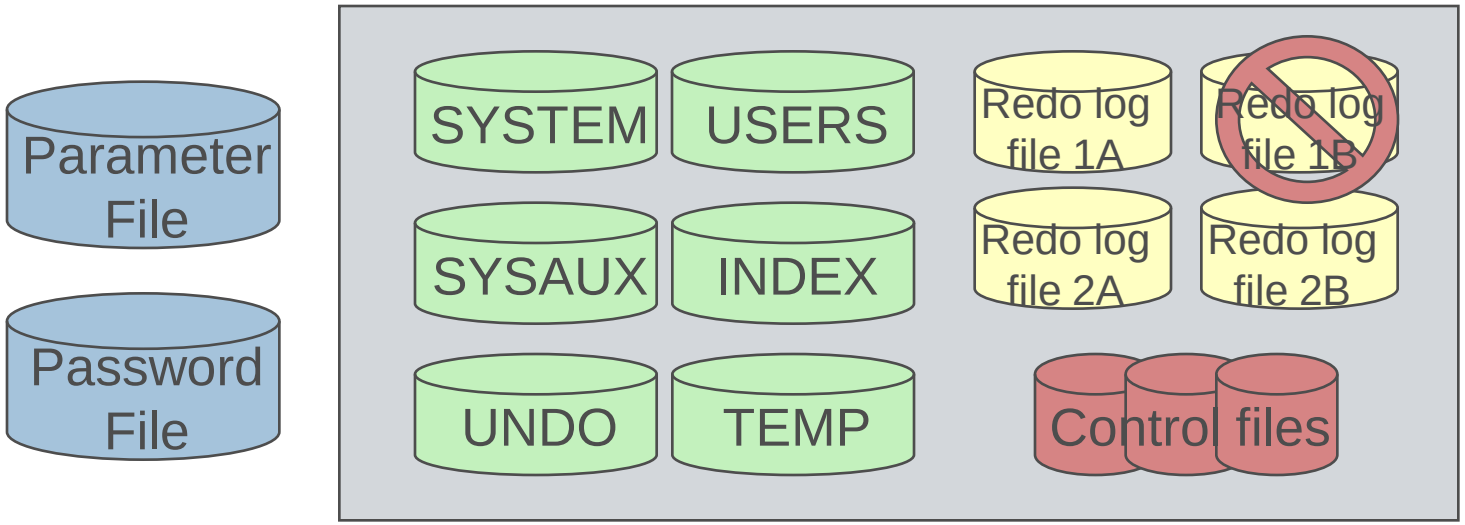
Creating New Temporary Tablespaces




```
SQL> CREATE TEMPORARY TABLESPACE temp2
  2> TEMPFILE '/oradata/temp2_01.tmp'
  3> SIZE 25M;
```

Recover a redo log group member

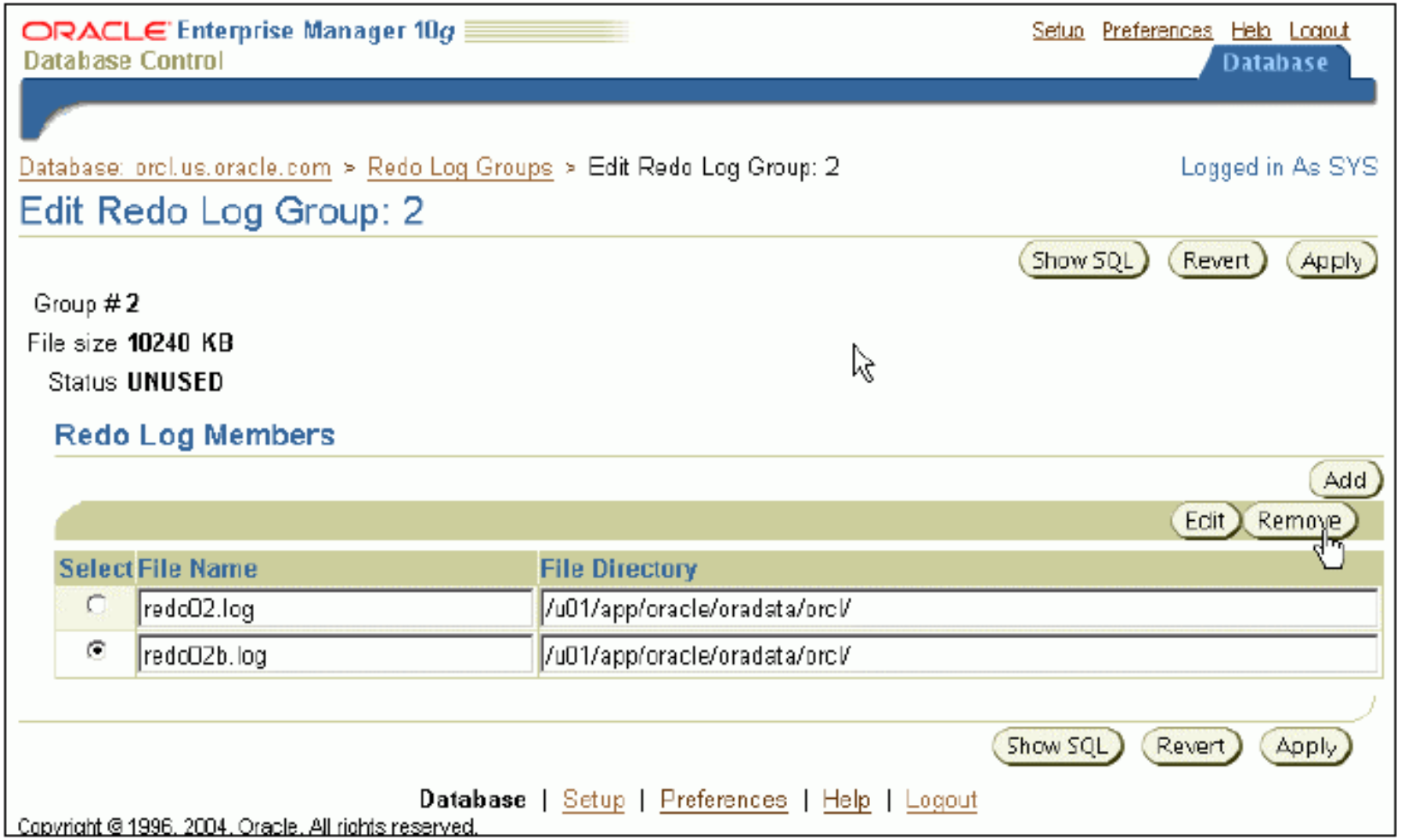
Re-creating Redo Log Files




```
SQL> ALTER DATABASE DROP LOGFILE MEMBER
  2> '/oradata/redo01b.log';
SQL> !rm /oradata/redo01b.log
SQL> ALTER DATABASE ADD LOGFILE MEMBER
  2> '/oradata/redo01b.log'
  3> TO GROUP 1;
```

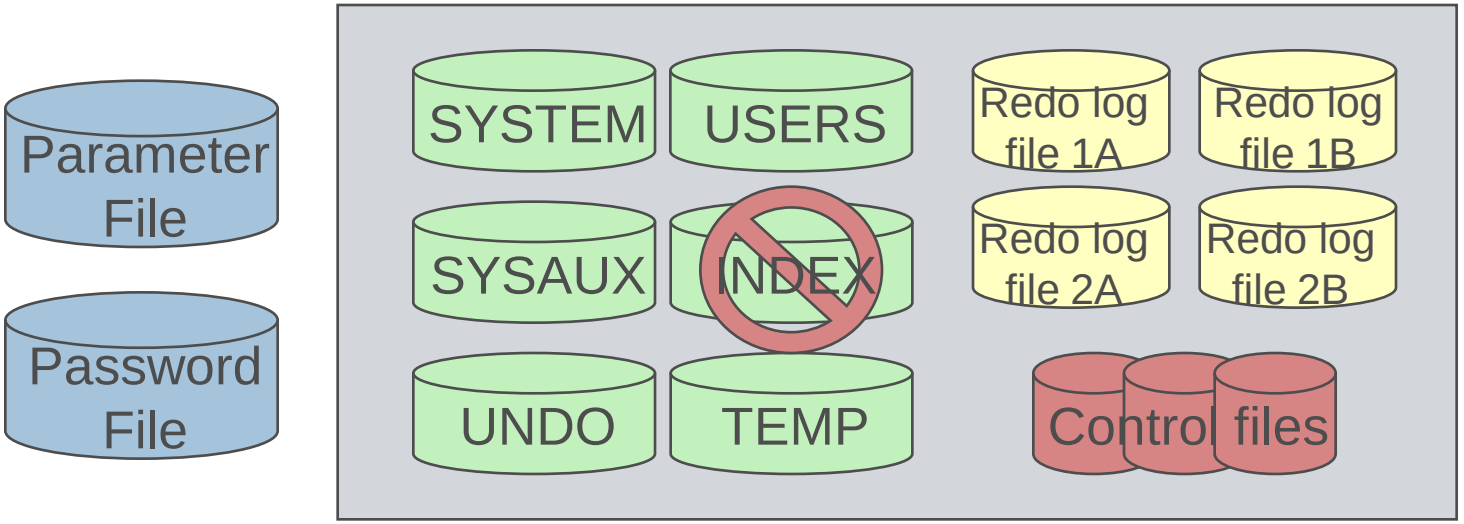
Recover a redo log group member

Re-creating Redo Log Files



Recover index tablespaces

Recovering an Index Tablespace

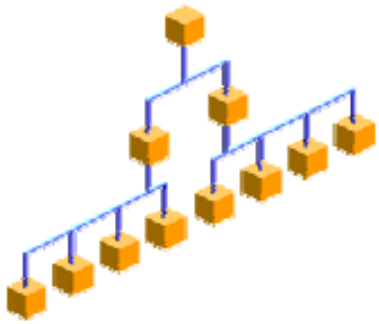


Recover index tablespaces

Re-creating Indexes

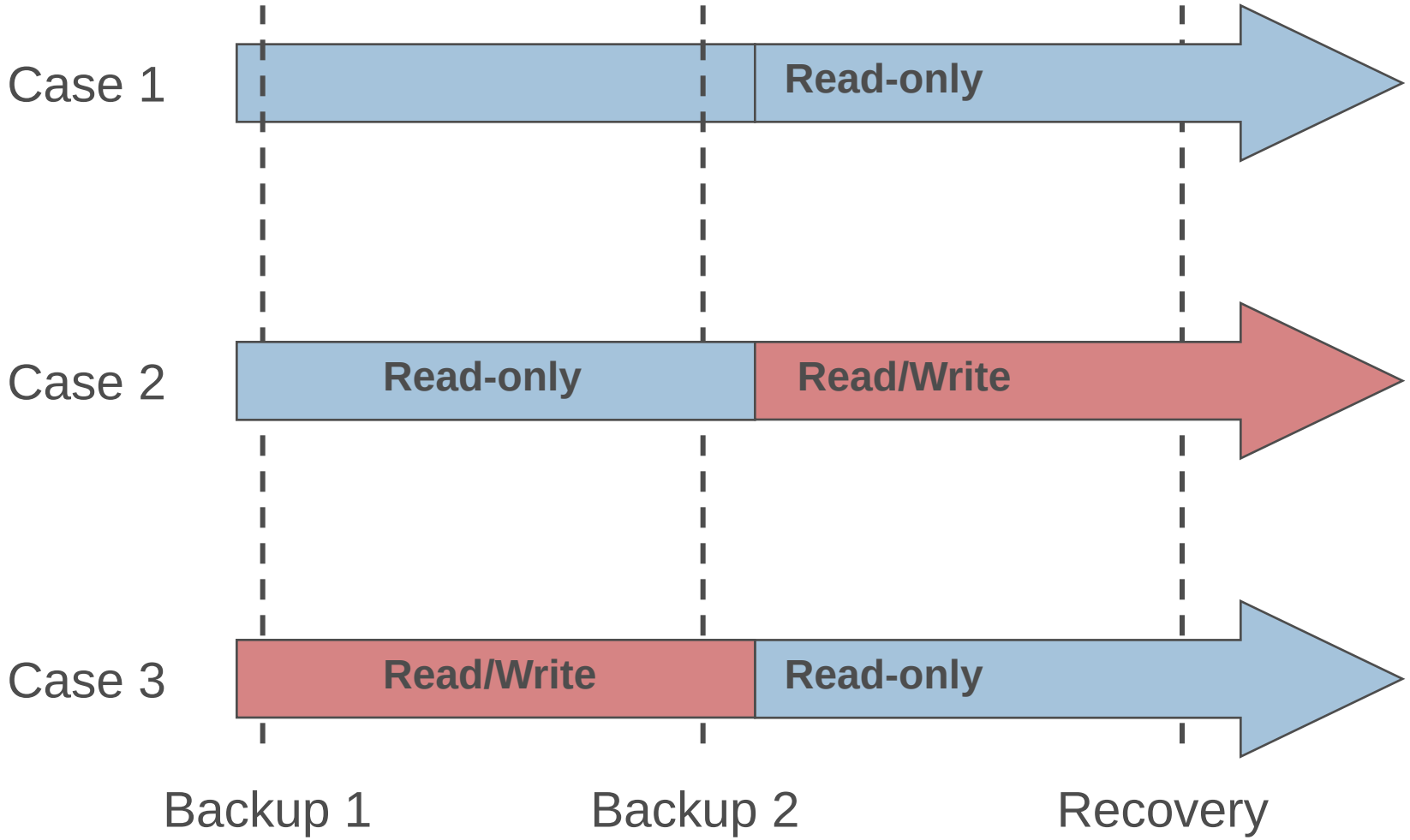
- Use options to reduce the time it takes to create the index:
  - **PARALLEL**
  - **NOLOGGING**

```
SQL> CREATE INDEX rname_idx
2 ON hr.regions (region_name)
3 PARALLEL 4;
```



Recover read-only tablespaces

Read-only Tablespace Recovery



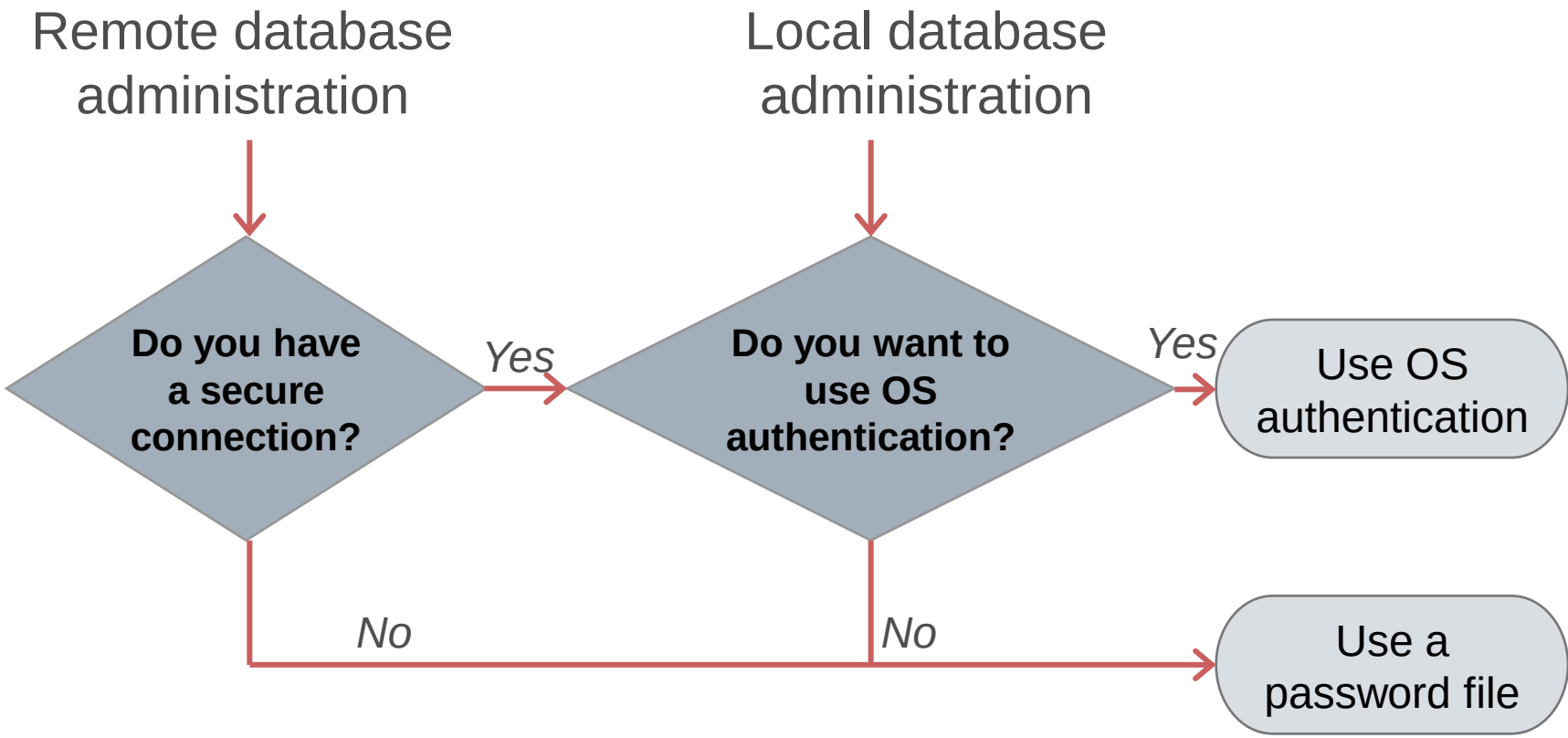
Recover read-only tablespaces

Read-only Tablespace Recovery Issues

- Special considerations must be taken for read-only tablespaces when:
  - Re-creating a control file
  - Renaming data files
  - Using a backup control file

Re-create the password file

Authentication Methods for Database Administrators



Re-create the password file

Loss of Password Authentication File

1. Log in to the database using OS authentication.
2. Set the **REMOTE_LOGIN_PASSWORDFILE** parameter to **NONE** and restart the database
3. Re-create the password file using **orapwd**.

```
$ orapwd file=$ORACLE_HOME/dbs/orapwORCL
password=admin entries=5
```

4. Set **REMOTE_LOGIN_PASSWORDFILE** to **EXCLUSIVE**.
5. Add users to the password file and assign appropriate privileges to each user.
6. Restart the instance.

Part 1 Summary

- Recover temporary tablespaces
- Recover a redo log group member
- Recover index tablespaces
- Recover read-only tablespaces
- Re-create the password file

Part 1 Stop-and-think

Do you have any questions ?

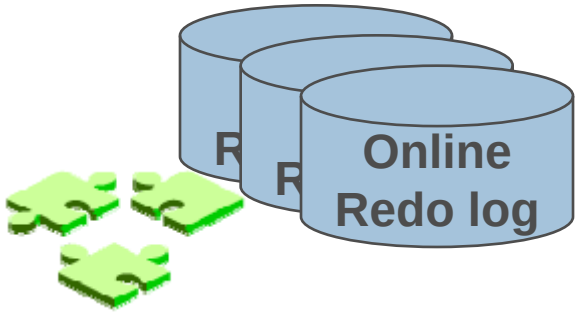
Database Recovery

Preview

- Recover the control file
- Explain reasons for incomplete recovery
- Describe incomplete recovery methodology
- Recover the database to a specific point in time using

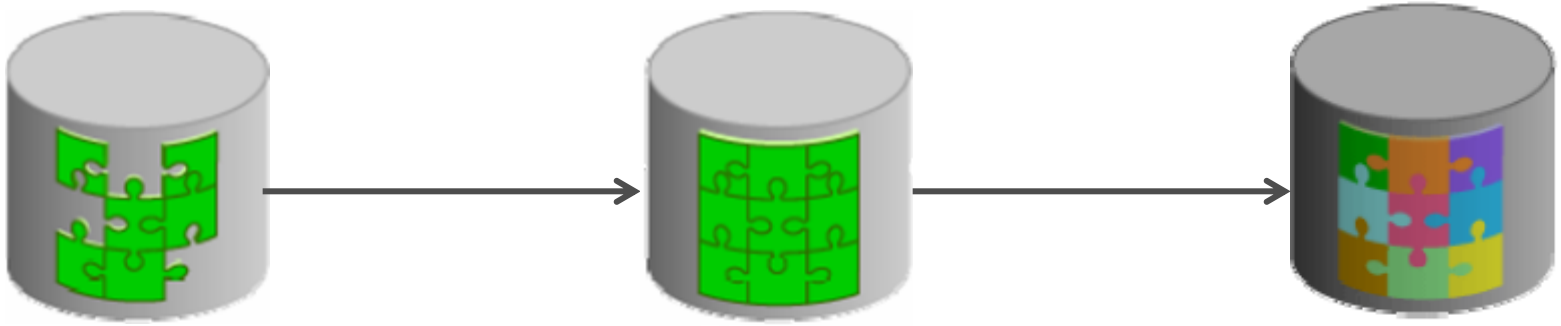
Recover the control file

Recovery Steps



Changes applied

Undo applied



Restored data files

Data files containing committed and uncommitted transactions

Recovered data files

Recover the control file

Server Managed Recovery: RESTORE and RECOVER Commands

```
run{
sql "ALTER TABLESPACE indx_tbs OFFLINE IMMEDIATE";
   RESTORE TABLESPACE indx_tbs;
   RECOVER TABLESPACE indx_tbs DELETE ARCHIVELOG;
   sql "ALTER TABLESPACE indx_tbs ONLINE";
}
```

Recover the control file

User-Managed Recovery Procedures: `RECOVER` Command

- Restore all database files from backup and recover the database:

```
SQL> RECOVER DATABASE
```

- Restore the damaged data files from a backup and recover the data files:

```
SQL> RECOVER TABLESPACE index_tbs
```

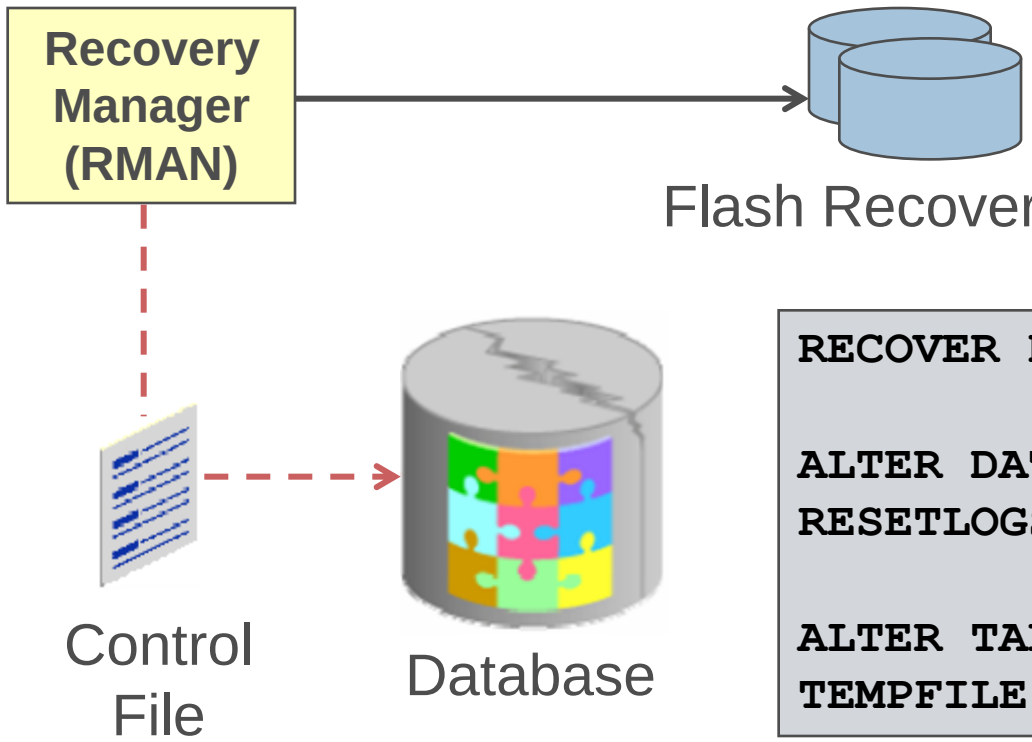
- or

```
SQL> RECOVER DATAFILE
2> '/oradata/indx01.dbf'
```

Recover the control file

Recovering a Control File Autobackup

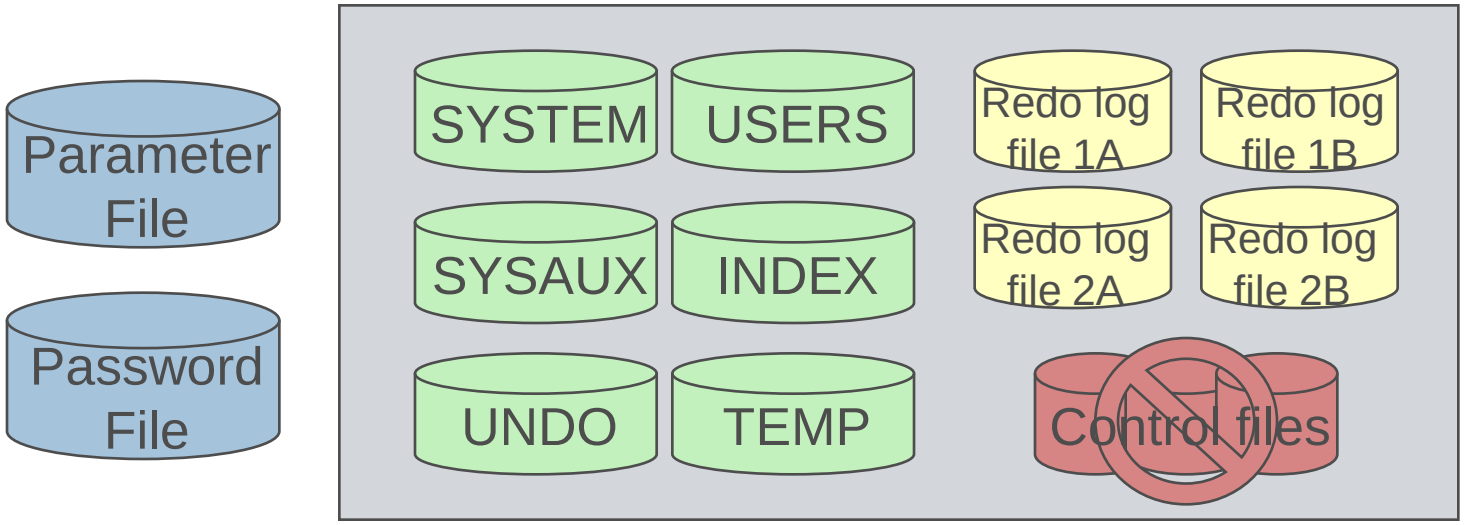
```
RMAN> RESTORE CONTROLFILE TO
2> '/oradata/spfile.bak' FROM AUTOBACKUP;
```




```
RECOVER DATABASE;

ALTER DATABASE OPEN
RESETLOGS;

ALTER TABLESPACE temp ADD
TEMPFILE...;
```

Recover the control file

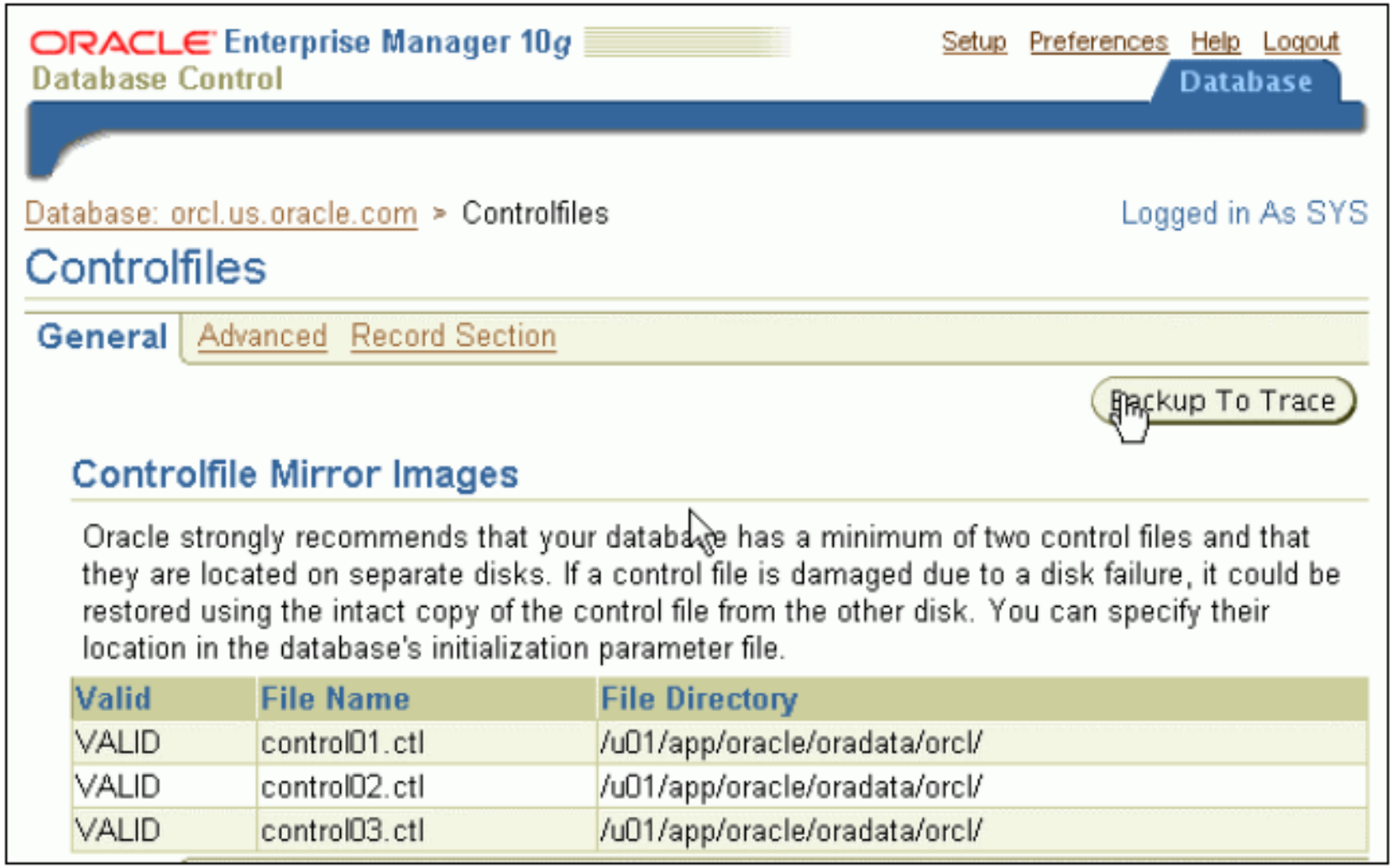
Creating a New Control File




```
SQL> ALTER DATABASE BACKUP CONTROLFILE
     TO TRACE;
```

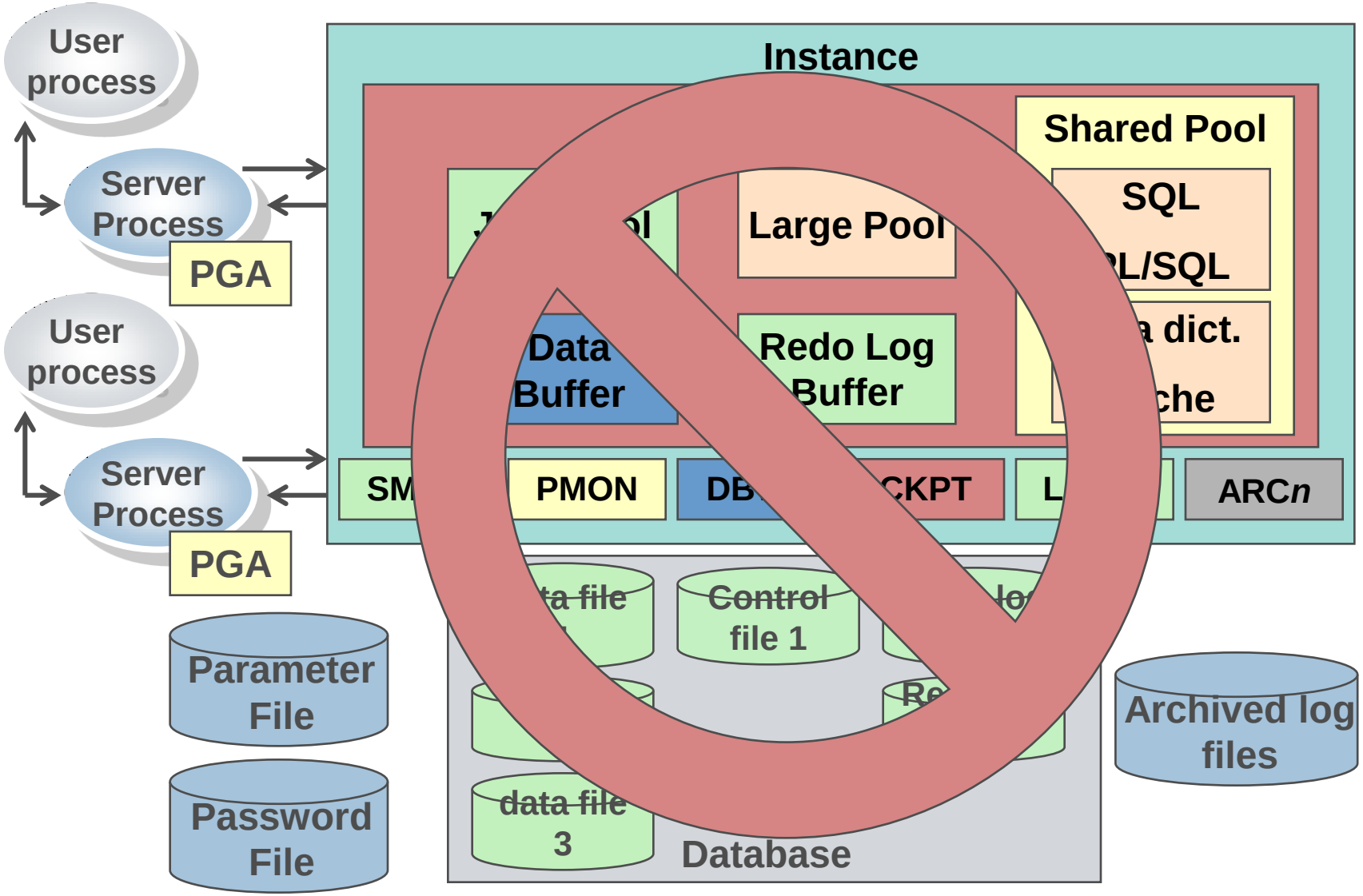
Recover the control file

Creating a New Control File



Explain reasons for incomplete recovery

Recovery Manager Features



Explain reasons for incomplete recovery

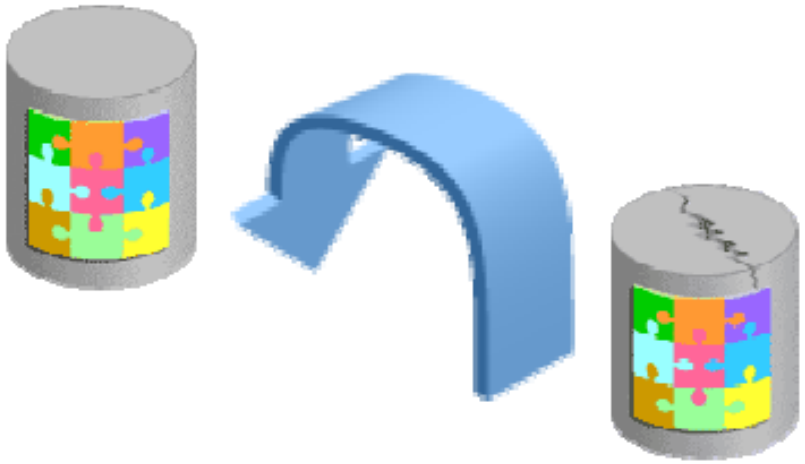
Situations Requiring Incomplete Recovery

- Complete recovery fails because of a missing archived log file
- One or more unarchived redo log files and a data file are lost
- A backup of the control file is used to open or recover the database

Describe incomplete recovery methodology

Types of Incomplete Recovery

- There are four types of incomplete recovery:
  - Time-based recovery
  - Cancel-based recovery
  - Change-based recovery
  - Log sequence recovery



Describe incomplete recovery methodology

Incomplete Recovery Best Practices

- Follow all steps carefully.
- Take whole database backups before and after recovery.
- Always verify that the recovery was successful.
- Back up and remove archived logs.

Describe incomplete recovery methodology

Using RECOVER for Incomplete Recovery

- Recover a database until time:

```
SQL> RECOVER DATABASE UNTIL
   2 TIME '2003-12-14 12:10:03';
```

- Recover a database until cancel:

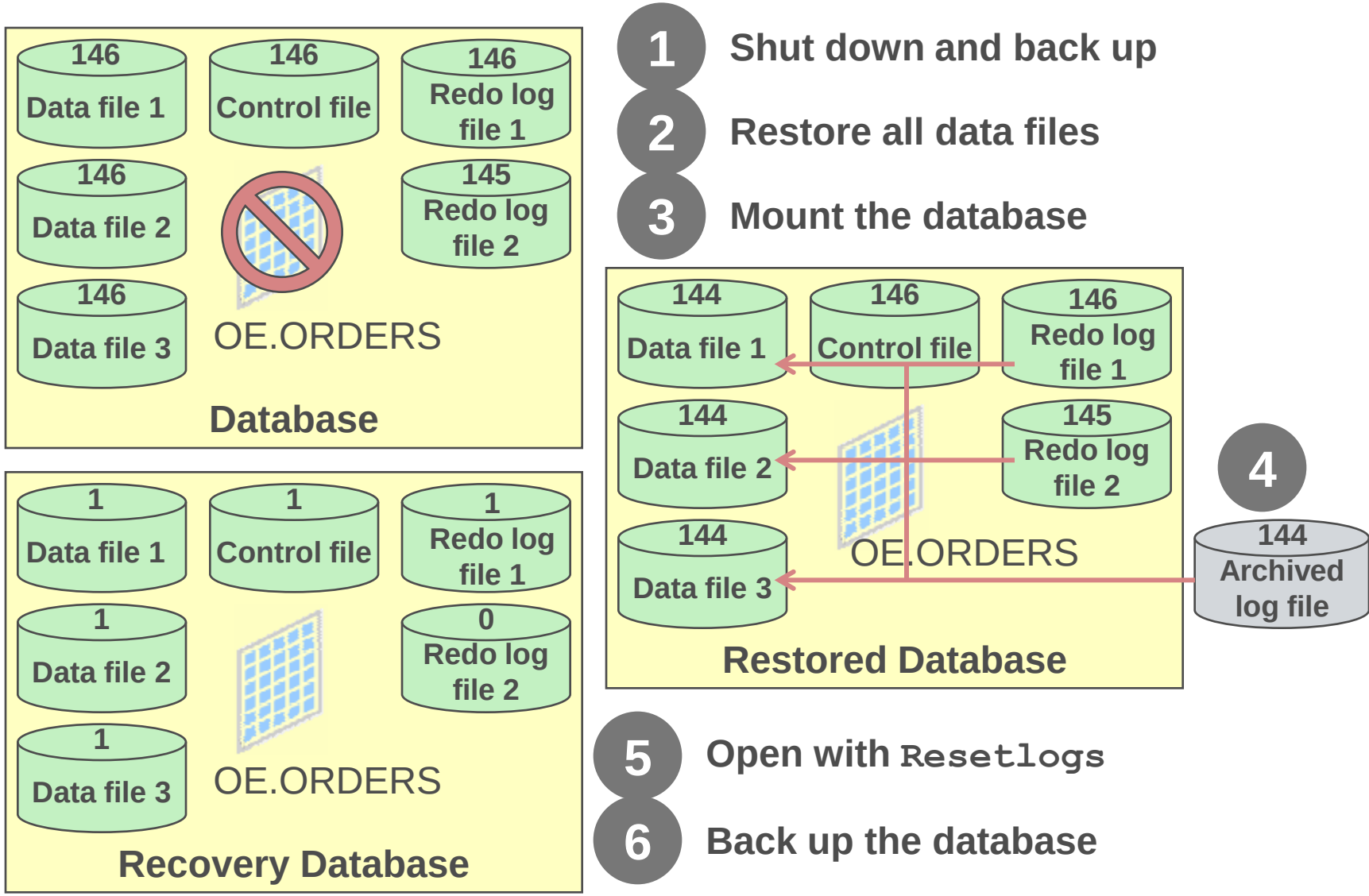
```
SQL> RECOVER DATABASE UNTIL CANCEL;
```

- Recover using backup control file:

```
SQL> RECOVER DATABASE
2 UNTIL TIME '2003-12-14 12:10:03'
3 USING BACKUP CONTROLFILE;
```

Describe incomplete recovery methodology

UNTIL TIME Recovery Example



Recover the database to a specific point in time using

`UNTIL TIME` Recovery Steps

1. Shutdown database
2. Restore data files
3. Mount the database
4. Recover the database
5. Open database with **`RESETLOGS`** option
6. Backup the database

```
SQL> shutdown immediate
$ cp /BACKUP/* /u01/db01/ORADATA
SQL> startup mount
SQL> recover database until time '2004-05-
28:11:44:00';
SQL> alter database open resetlogs;
SQL> shutdown;
$ cp /u01/db01/ORADATA/* /BACKUP
```

Recover the database to a specific point in time using

Cancel-Based Recovery: Example

- Scenario:
  - The current time is 12:00 p.m. on May 28, 2004.
  - The **ORDERS** table was dropped while someone was trying to fix corrupted data blocks.
  - Log files exist on the same disk as the data files.
  - The table was dropped at approximately 11:45 a.m.
  - Staff are currently in a meeting.

Recover the database to a specific point in time using

Cancel-Based Recovery: Example

- Findings:
  - Redo logs are not multiplexed.
  - One of the online redo logs is missing.
  - The missing redo log is not archived.
  - The redo log contained information from 11:34 a.m.
  - Twenty-six minutes of data will be lost.
  - Users can recover their data.

Recover the database to a specific point in time using

Incomplete Recovery and the Alert Log

- Check the alert log before and after recovery
- Look for error information, hints, and SCNs
- Confirm steps in the recovery process were successful

Recover the database to a specific point in time using

Incomplete Recovery of a Database Using RMAN

1. Mount the database.
2. Allocate multiple channels for parallelization.
3. Restore all data files.
4. Recover the database by using **`UNTIL TIME`**, **`UNTIL SEQUENCE`**, or **`UNTIL SCN`**.
5. Open the database by using **`RESETLOGS`**.
6. Perform a whole database backup.

Recover the database to a specific point in time using

RMAN Incomplete Recovery `UNTIL TIME`: Example

```
RMAN> RUN {
   2> SET UNTIL TIME = '2004-05-28 11:44:00';
   3> RESTORE DATABASE;
   4> RECOVER DATABASE;
   5> ALTER DATABASE OPEN RESETLOGS; }
```

Recover the database to a specific point in time using

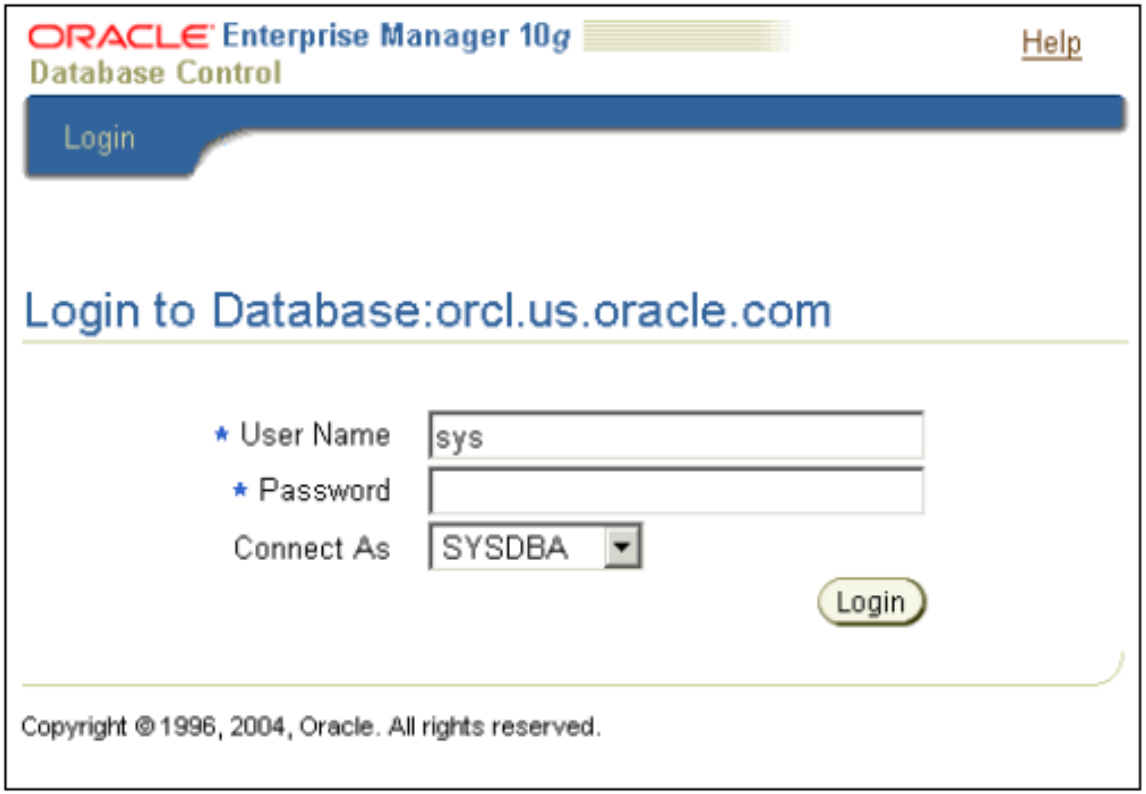
RMAN Incomplete Recovery UNTIL SEQUENCE: Example

```
RMAN> RUN {
   2> SET UNTIL SEQUENCE 120 THREAD 1;
   3> ALTER DATABASE MOUNT;
   4> RESTORE DATABASE;
   5> RECOVER DATABASE; # recovers through log
119
   6> ALTER DATABASE OPEN RESESTLOGS;
   7> }
```

Recover the database to a specific point in time using

Recovery Using Enterprise Manager

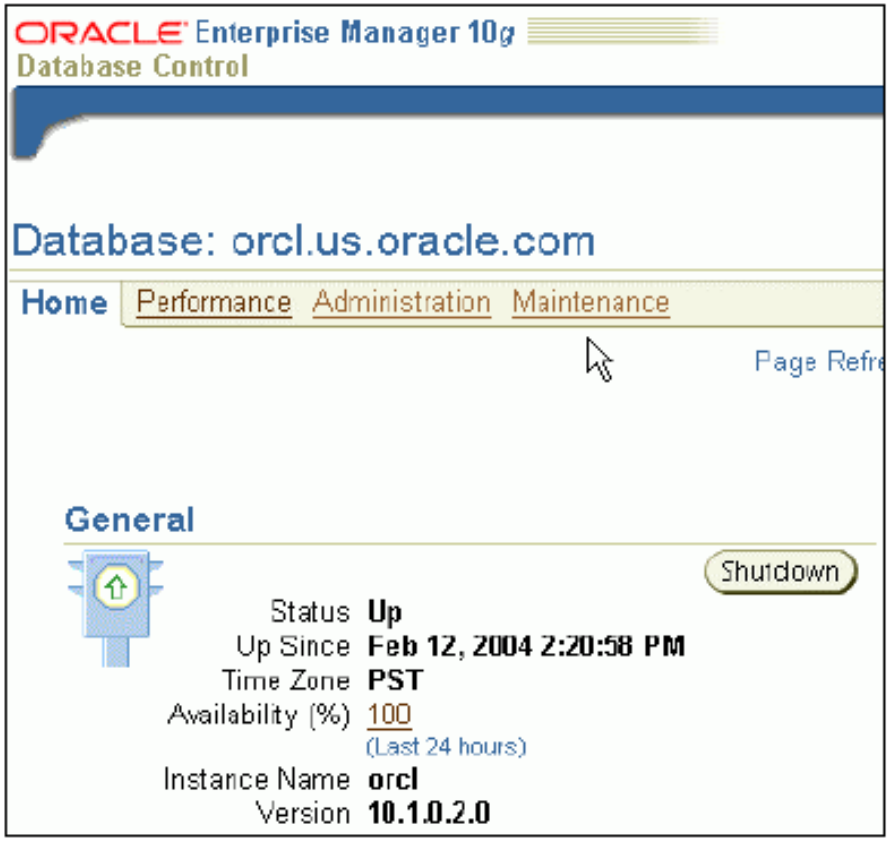
- Log in as a user with the **SYSDBA** privilege.



Recover the database to a specific point in time using

Recovery Using Enterprise Manager

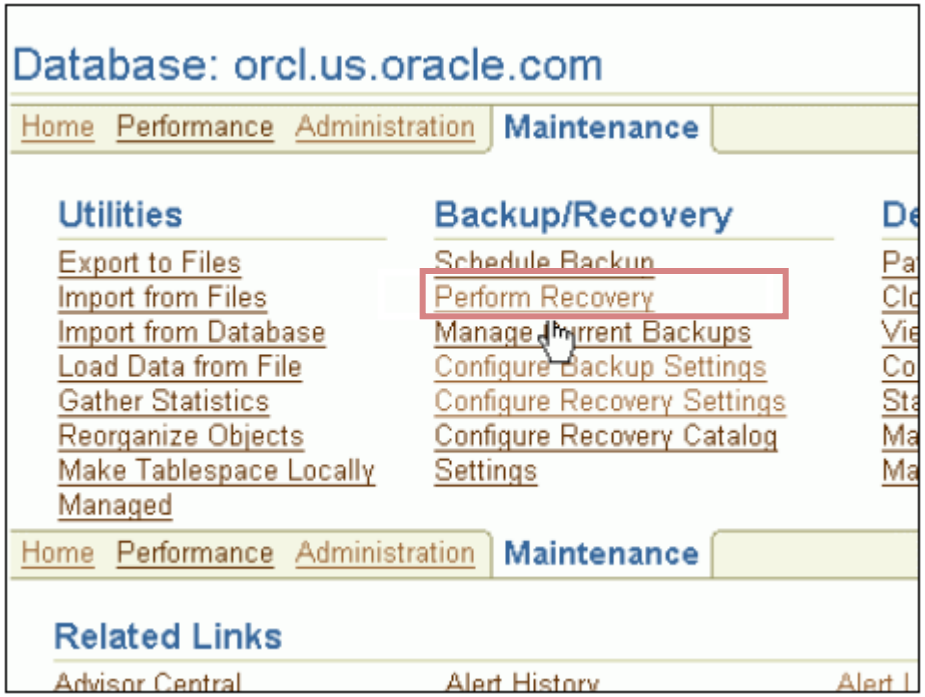
- Click on the Maintenance tab.



Recover the database to a specific point in time using

Recovery Using Enterprise Manager

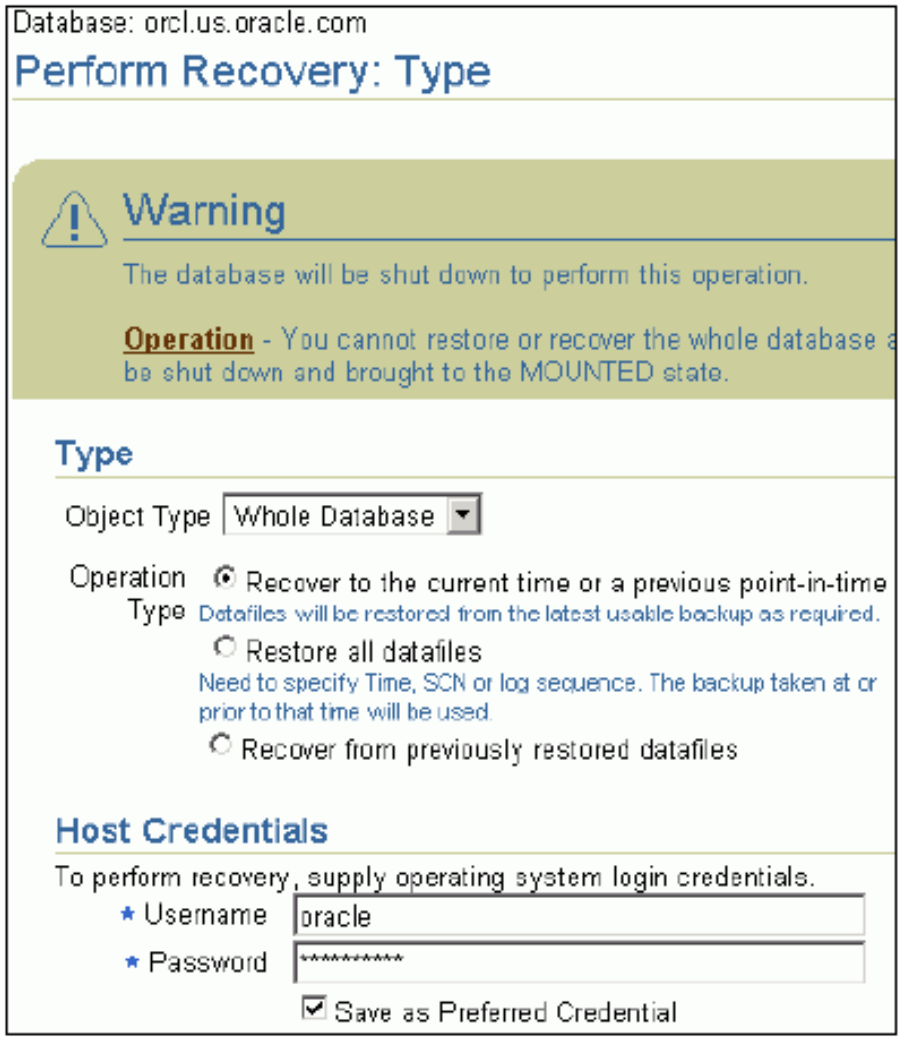
- Select Perform Recovery.



Recover the database to a specific point in time using

Recovery Using Enterprise Manager

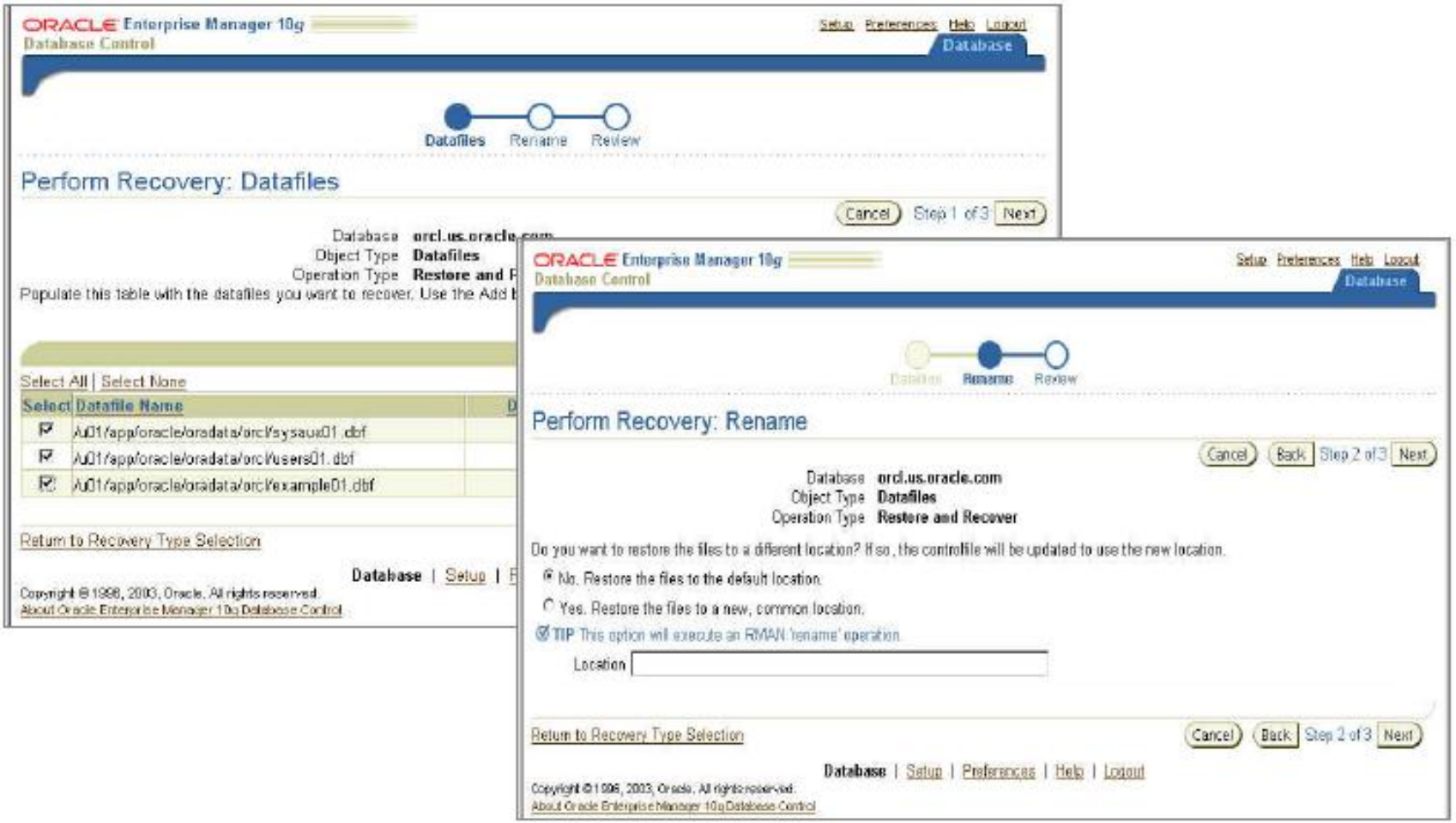
- Select Whole Database and enter OS login credentials.



Recover the database to a specific point in time using

Recovery Using Enterprise Manager

- Select data files.



Recover the database to a specific point in time using

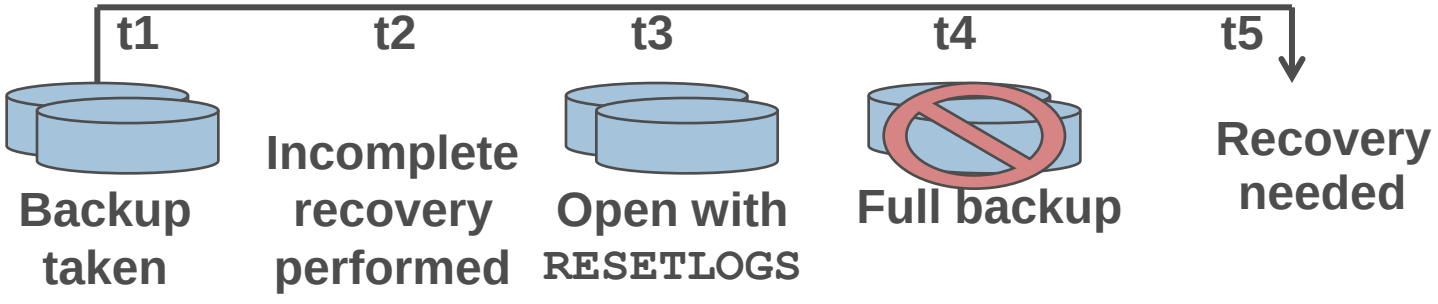
Recovery Using Enterprise Manager

- Final review

Recover the database to a specific point in time using

Simplified Recovery Through RESETLOGS

- The database can be opened immediately after **RESETLOGS**. (No longer need to take a full backup)
- No changes are required to existing scripts.
- Recovery through **RESETLOGS** can be used for:
  - Incomplete recovery
  - Recovery using a backup control file



Recover the database to a specific point in time using

Recovery Through RESETLOGS: Changes

```
LOG_ARCHIVE_FORMAT="LOG%t_%s_%r.arc"
```

```
SQL> SELECT recid, thread#, sequence#,
  2         resetlogs_change#, resetlogs_time
  3 FROM v$log_history;
```

```
SQL> SELECT recid, file#,
  2         resetlogs_change#, resetlogs_time
  3 FROM v$offline_range;
```

```
SQL> SELECT incarnation#, resetlogs_id,
  2         prior_incarnation#, status
  3 FROM v$database_incarnation;
```

Part 2 Summary

- Recover the control file
- Explain reasons for incomplete recovery
- Describe incomplete recovery methodology
- Recover the database to a specific point in time using

Part 2 Stop-and-think

Do you have any questions ?

Flashback Database

Preview

- Describe Flashback Database architecture
- Configuring Flashback Database
- Monitor the Flashback Database
- Use the Enterprise Manager Recovery Wizard

Describe Flashback Database architecture

Flashback Technology Benefits

- Flashback technology is a revolutionary advance in recovery
- Traditional recovery techniques are slow
  - Entire database or file has to be restored, not just the incorrect data
  - Every change in the database log must be examined
- Flashback is fast
  - Changes are indexed by row and by transaction
  - Only the changed data is restored
- Flashback commands are easy
  - No complex multi-step procedures

Describe Flashback Database architecture

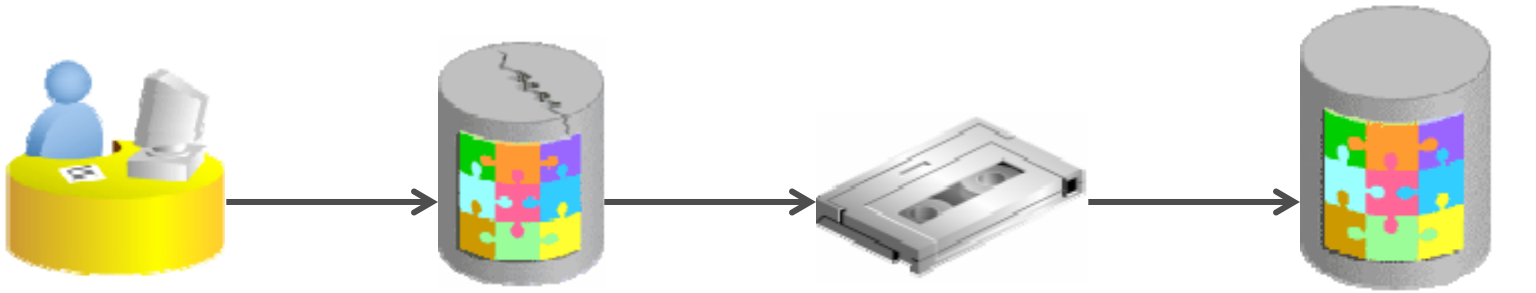
When to Use Flashback Technology

Object level	Scenario	Flashback Technology
Database	Drop User	Flashback Database
	Truncate Table	Flashback Database
	Batch job : partial changes	Flashback Database
Table	Drop table	Flashback Drop
	Update with wrong WHERE clause	Flashback Table
	Comparing current data against the data at some time in the past	Flashback Query
Tx	Batch Job runs twice, but not really sure of the objects affected	Flashback Query

Describe Flashback Database architecture

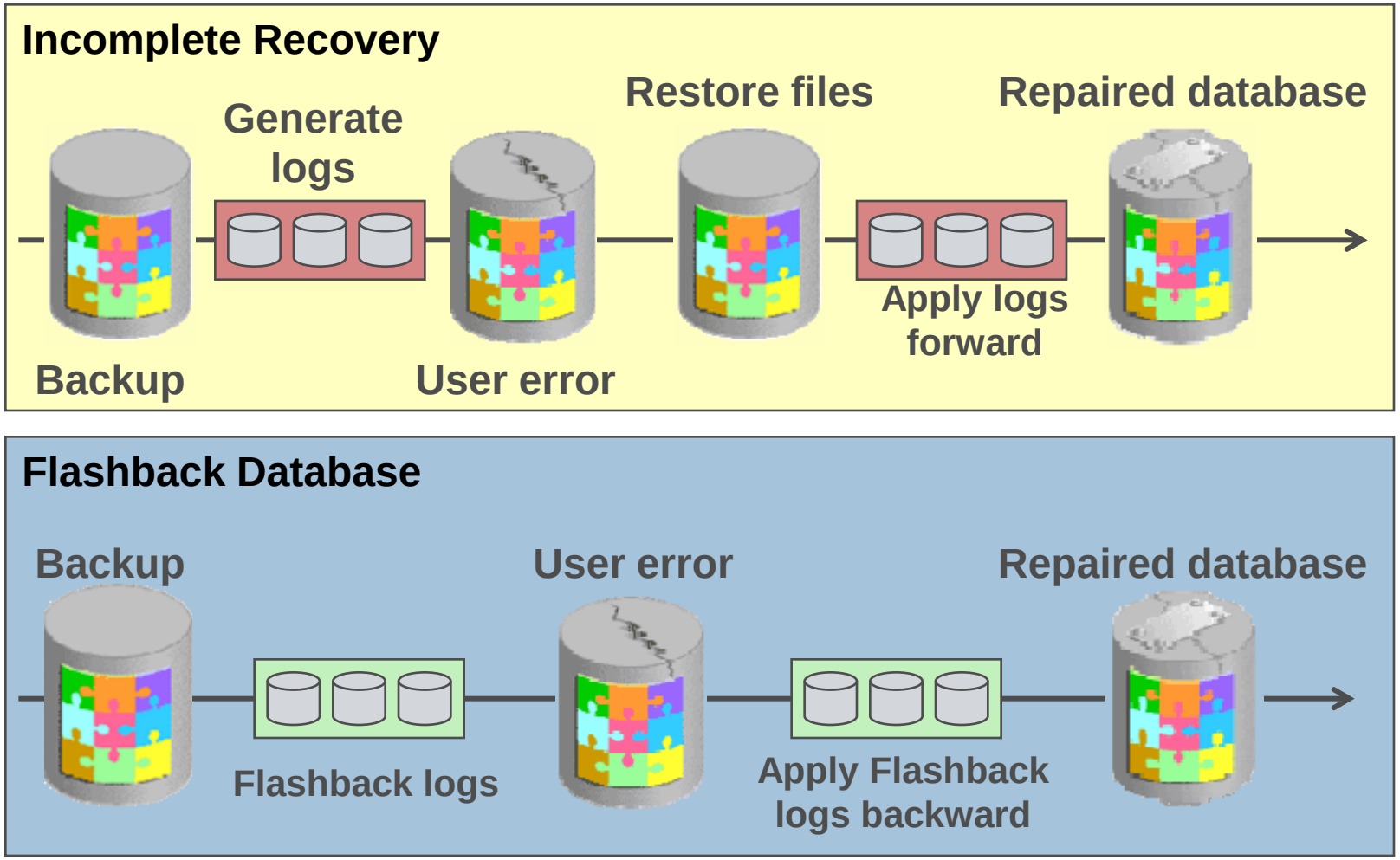
Flashback Database Overview

- The Flashback Database operation:
  - Works like a rewind button for the database.
  - Can be used in cases of logical data corruptions made by users.



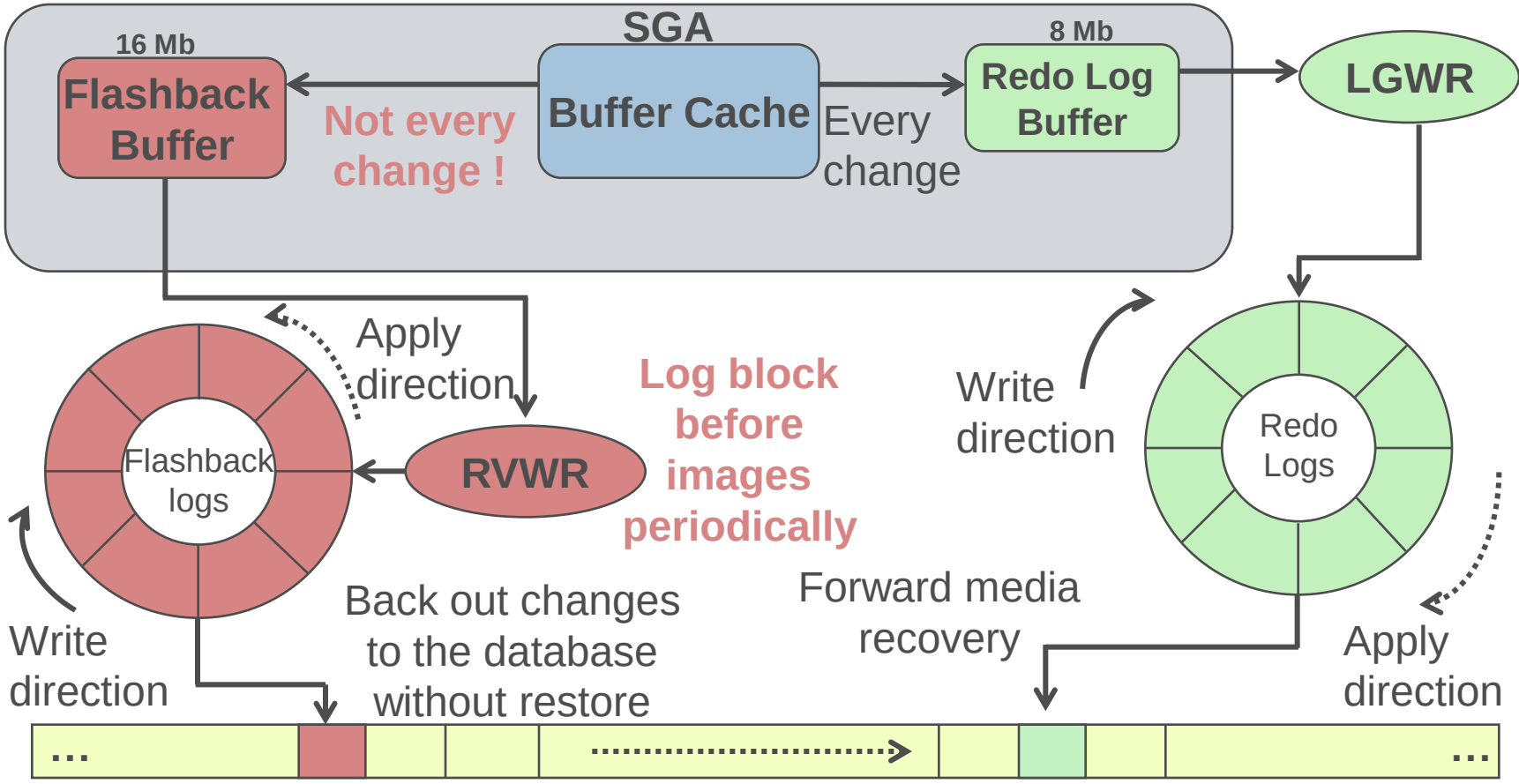
Describe Flashback Database architecture

Flashback Database Reduces Restore Time



Describe Flashback Database architecture

Recovery Manager Features



Describe Flashback Database architecture

Flashback Database Considerations

- When the Flashback Database operation completes, open the database:
  - In read-only mode to verify that the correct target time or SCN was used
  - With a **RESETLOGS** operation to allow for updates
- The opposite of flashback is recover
- You cannot use Flashback Database in the following situations:
  - The control file has been restored or re-created.
  - A tablespace has been dropped.
  - A data file has been shrunk.
  - You want to flashback before **RESETLOGS** operation.

Describe Flashback Database architecture

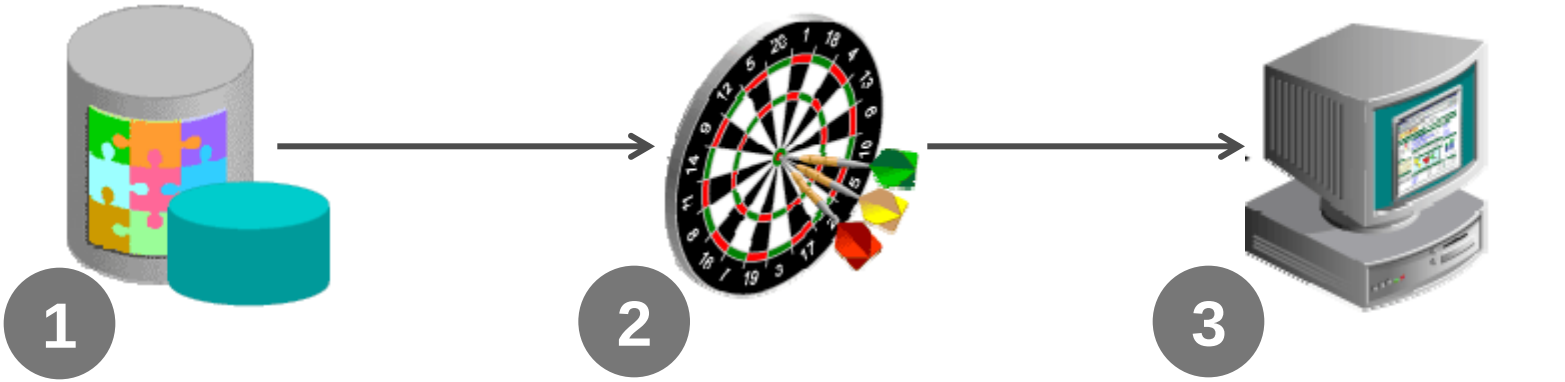
Flashback Database Examples

```
RMAN> FLASHBACK DATABASE TO TIME =
   2> TO_DATE('2004-05-27 16:00:00',
   3> 'YYYY-MM-DD HH24:MI:SS');

RMAN> FLASHBACK DATABASE TO SCN=23565;

RMAN> FLASHBACK DATABASE
   2> TO SEQUENCE=223 THREAD=1;
```

```
SQL> FLASHBACK DATABASE
   2 TO TIMESTAMP(SYSDATE-1/24);

SQL> FLASHBACK DATABASE TO SCN 53943;
```

Configuring Flashback Database

Configuring Flashback Database



1. Configure the flash recovery area
2. Set the retention target
3. Enable Flashback Database

```
SQL> ALTER SYSTEM SET
   2 DB_FLASHBACK_RETENTION_TARGET=2880
   3 SCOPE=BOTH;
```

```
SQL> ALTER DATABASE FLASHBACK ON;
```

Configuring Flashback Database

Best Practices for the Database and Flash Recovery Area

- Use the flash recovery area for recovery-related files:
  - Simplifies location of database backups
  - Automatically manages the disk space allocated for recovery files
  - Does not require changes to existing scripts
  - Puts database backups, archive logs, and control file backups in the flash recovery area

Configuring Flashback Database

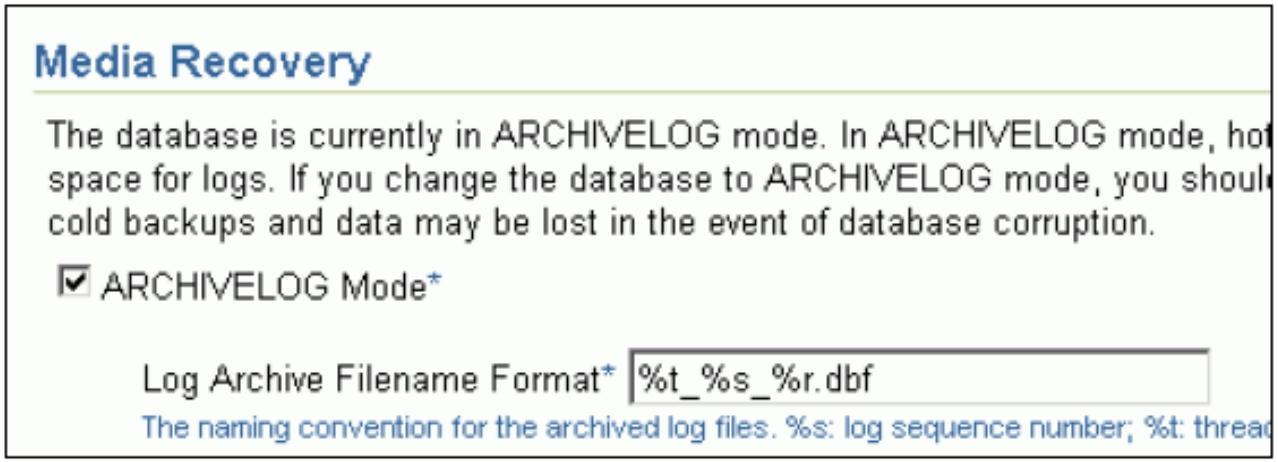
Flash Recovery Area Space Usage

- Configure the retention policy to the minimum value appropriate for your database
- Backup the archive log files regularly and delete the files upon completion of the backup
- Use the RMAN **`REPORT OBSOLETE`** and **`DELETE OBSOLETE`** commands to remove unneeded backups and file copies

Configuring Flashback Database

Configure Flashback Database with EM

- Make sure the database is in **ARCHIVELOG** mode.



Configuring Flashback Database

Excluding Tablespaces from Flashback Database

```
ALTER TABLESPACE <ts_name> FLASHBACK {ON|OFF}
```

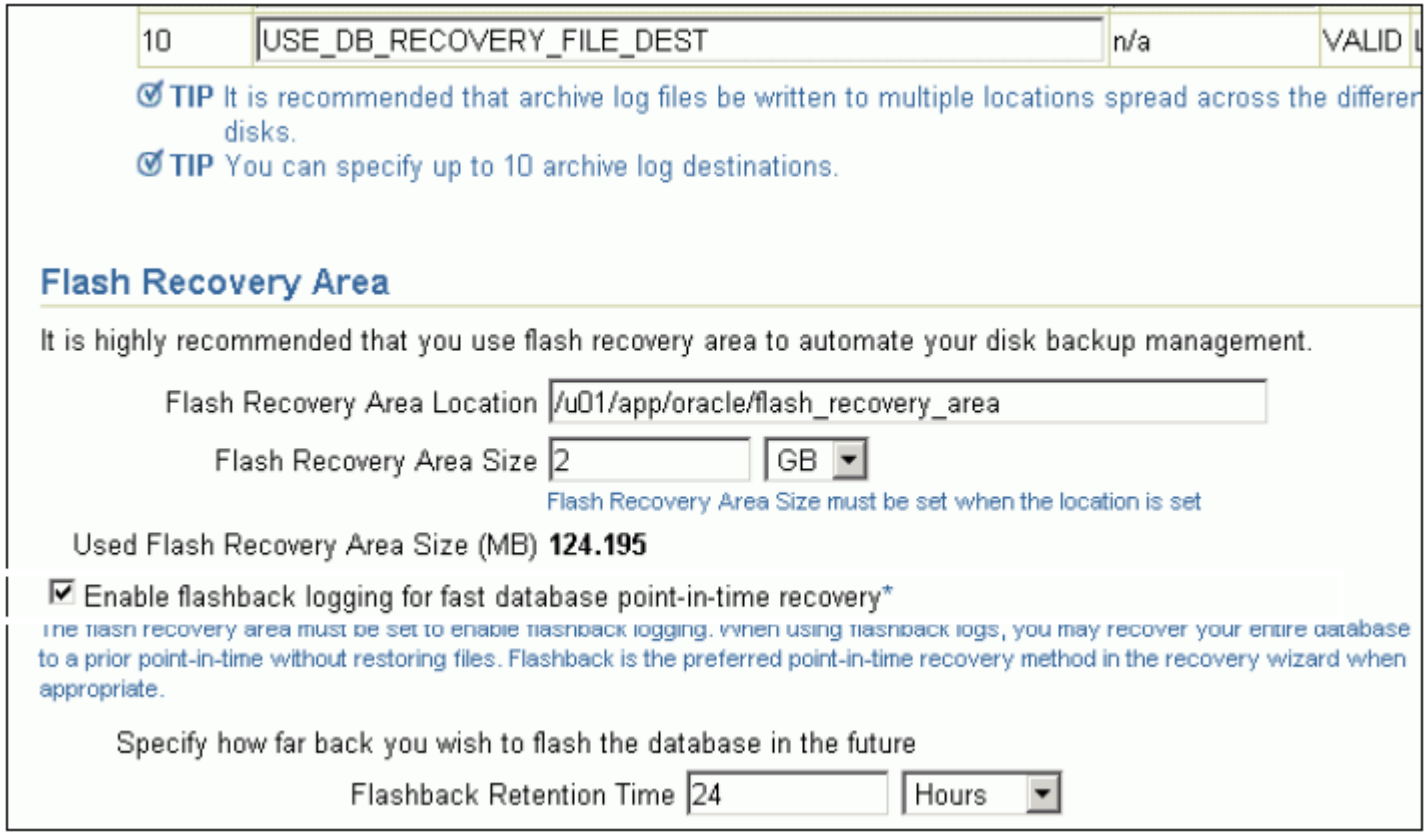
```
SQL> SELECT tablespace_name, flashback_on
   2 FROM v$tablespace;
```

- Take the tablespace offline before you perform the Flashback Database recovery.
- Drop the tablespace or recover the offline files with traditional point-in-time recovery.

Configuring Flashback Database

Configure Flashback Database with EM

- Enable Flashback logging and specify flash recovery area



Monitor the Flashback Database

Monitoring Flashback Database

- Adjust the flash recovery area disk quota:

```
SQL> SELECT estimated_flashback_size,
   2        flashback_size
   3 FROM   V$FLASHBACK_DATABASE_LOG;
```

- Determine the current flashback window:

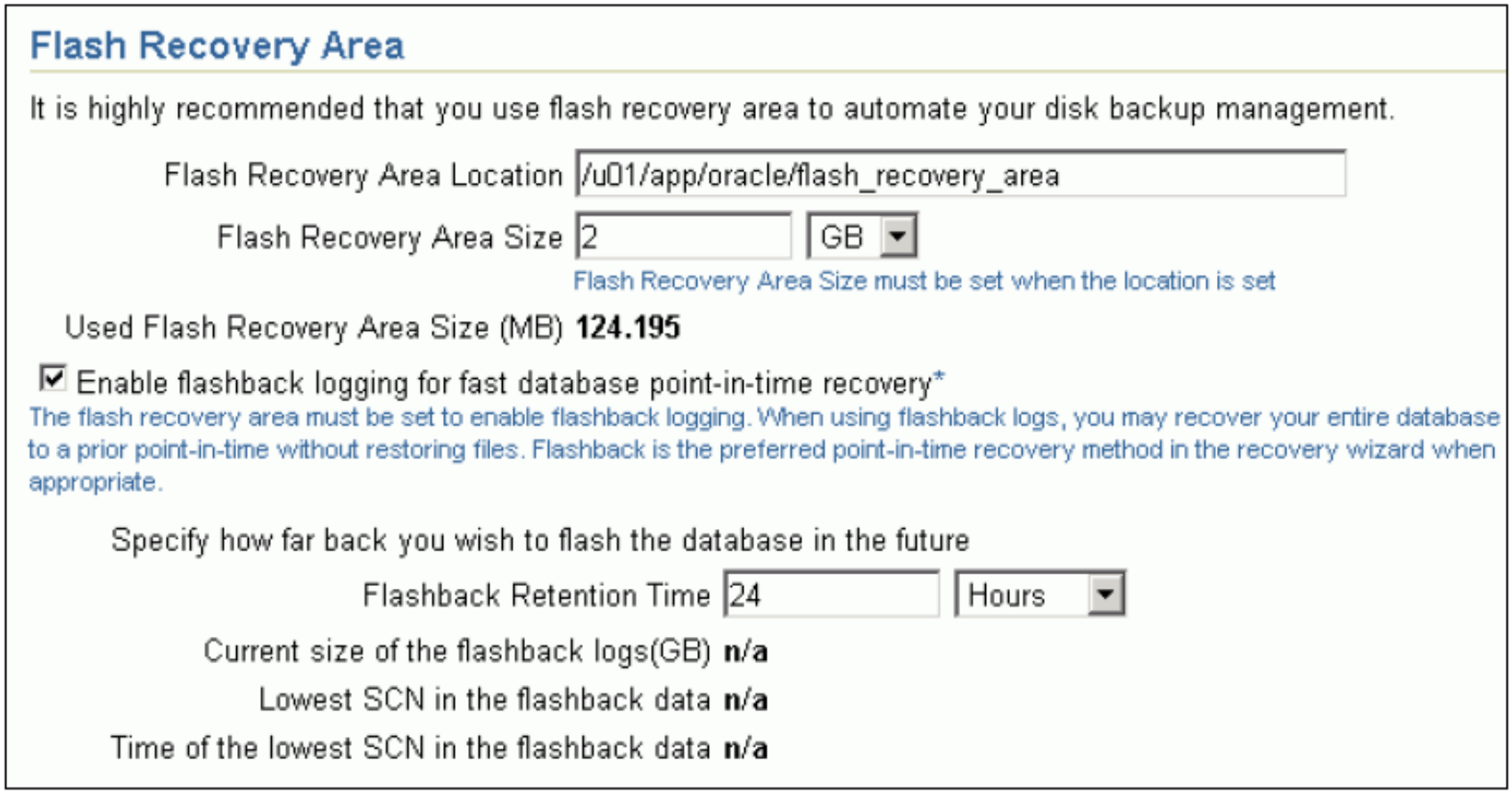
```
SQL> SELECT oldest_flashback_scn,
   2        oldest_flashback_time
   3 FROM   V$FLASHBACK_DATABASE_LOG;
```

- Monitor logging in the Flashback Database logs:

```
SQL> SELECT *
   2 FROM   V$FLASHBACK_DATABASE_STAT;
```

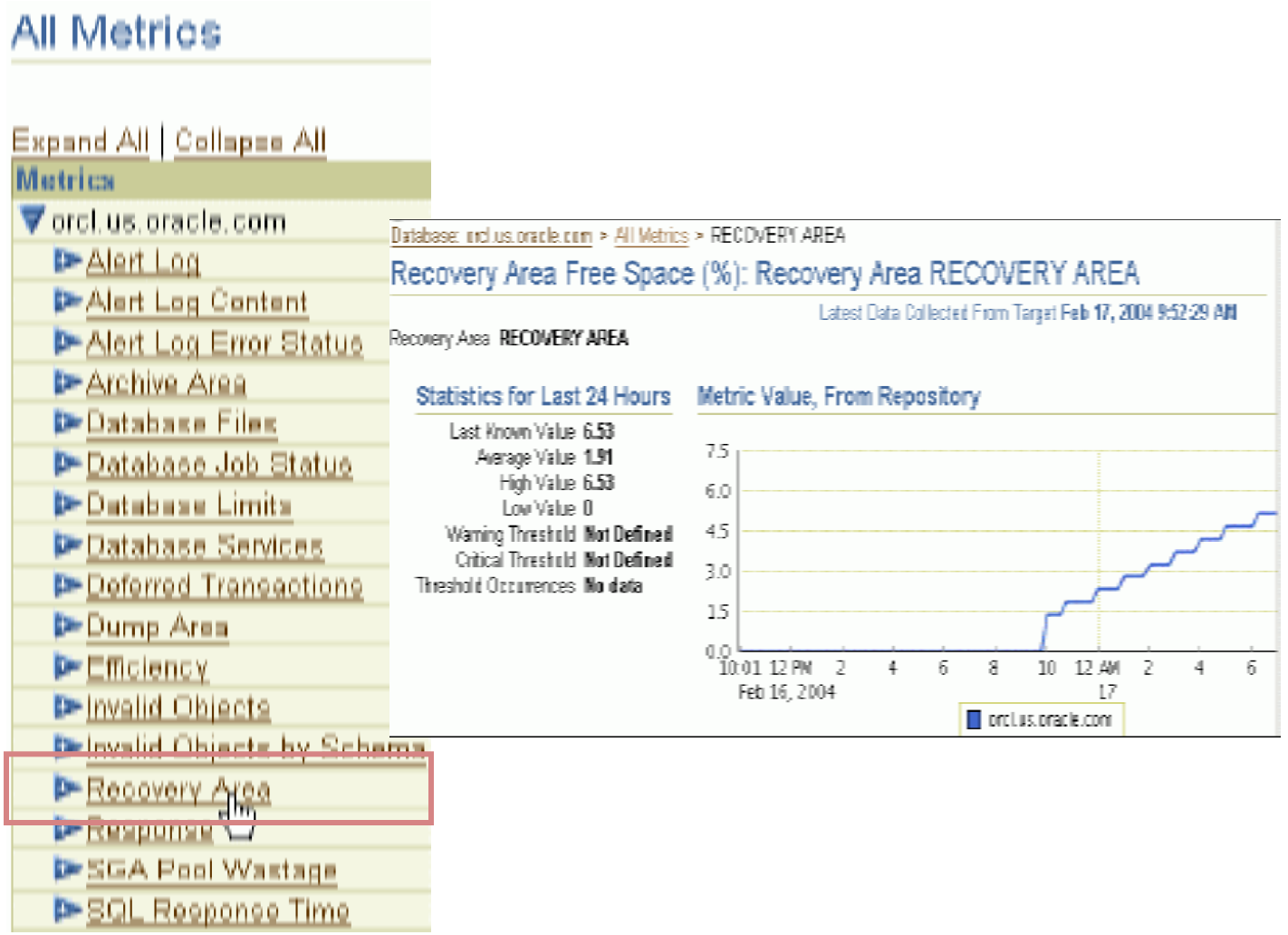
Monitor the Flashback Database

Monitoring Flashback Database with EM



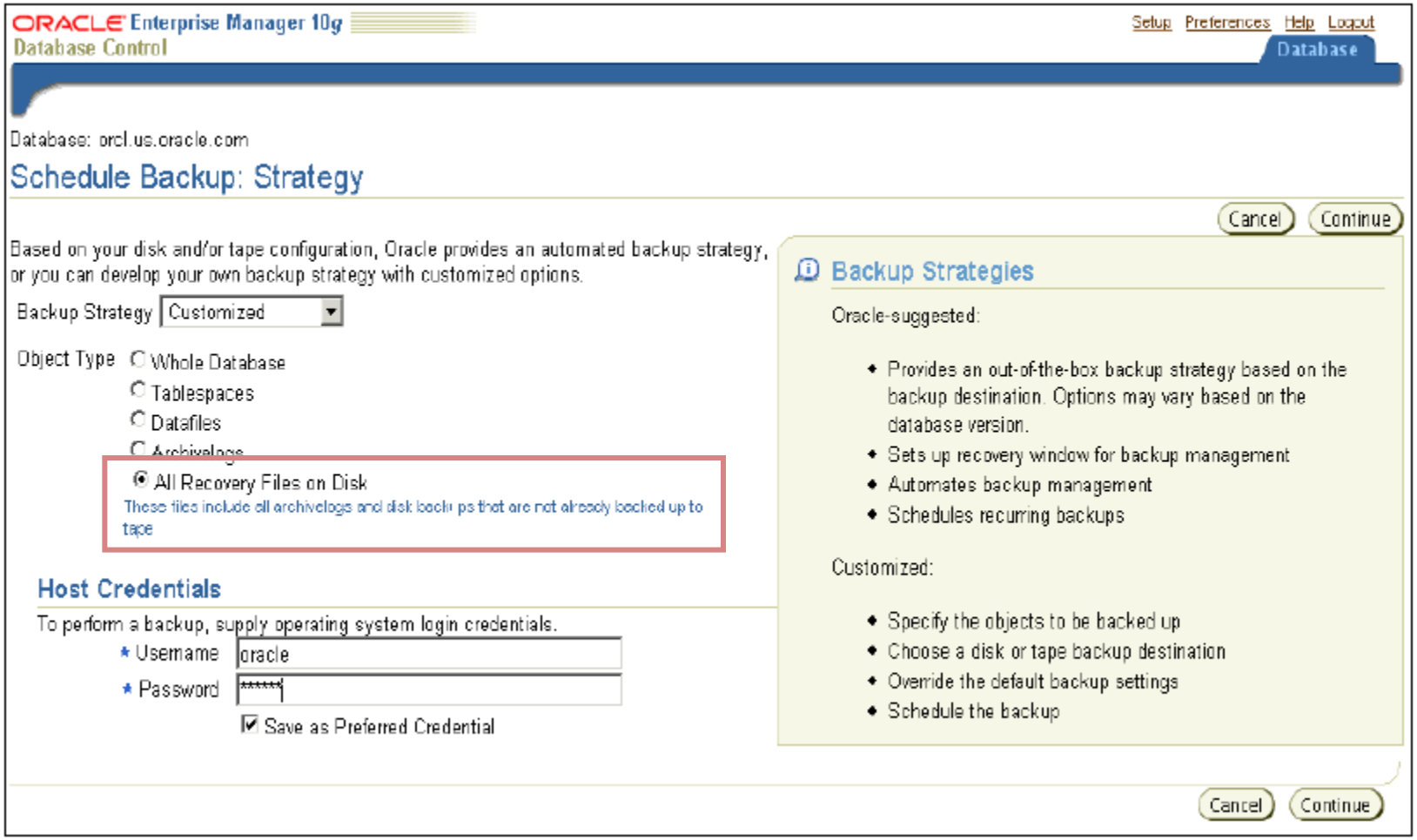
Monitor the Flashback Database

Monitoring Flash Recovery Area with EM



Use the Enterprise Manager Recovery Wizard

Backing Up the Flash Recovery Area

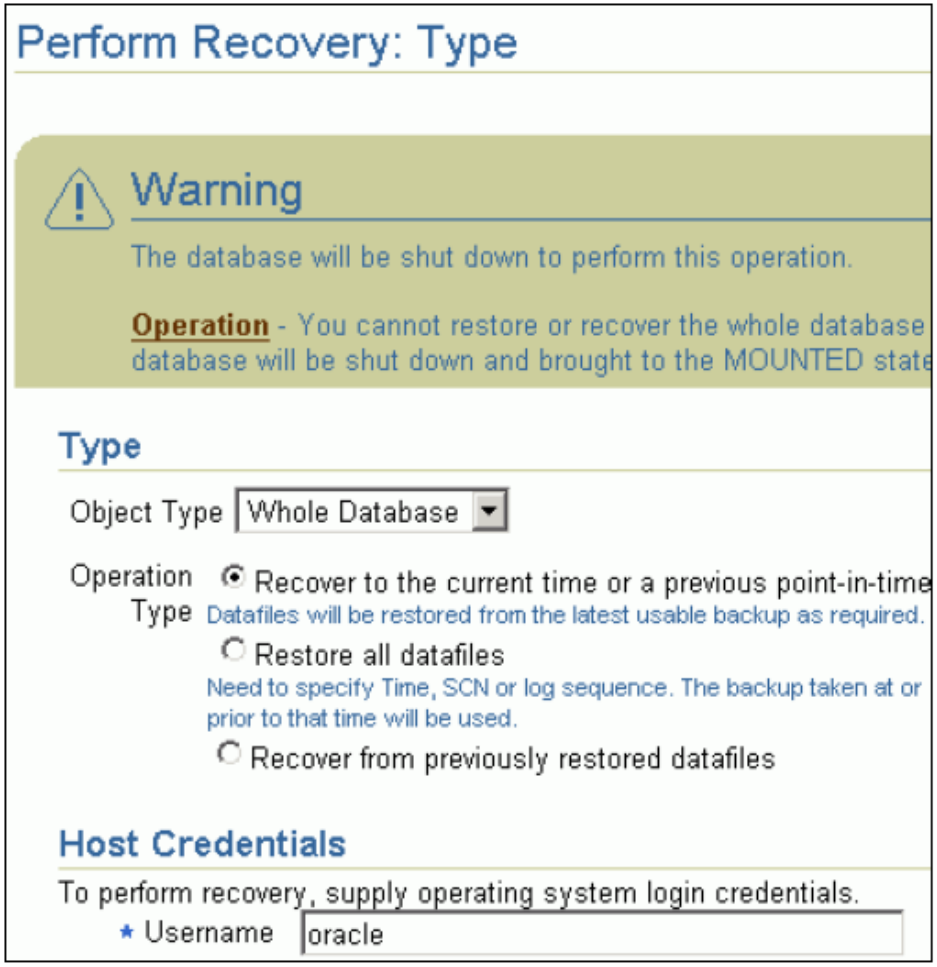



```
RMAN> BACKUP RECOVERY FILES;
```

Use the Enterprise Manager Recovery Wizard

Flashback Database with EM

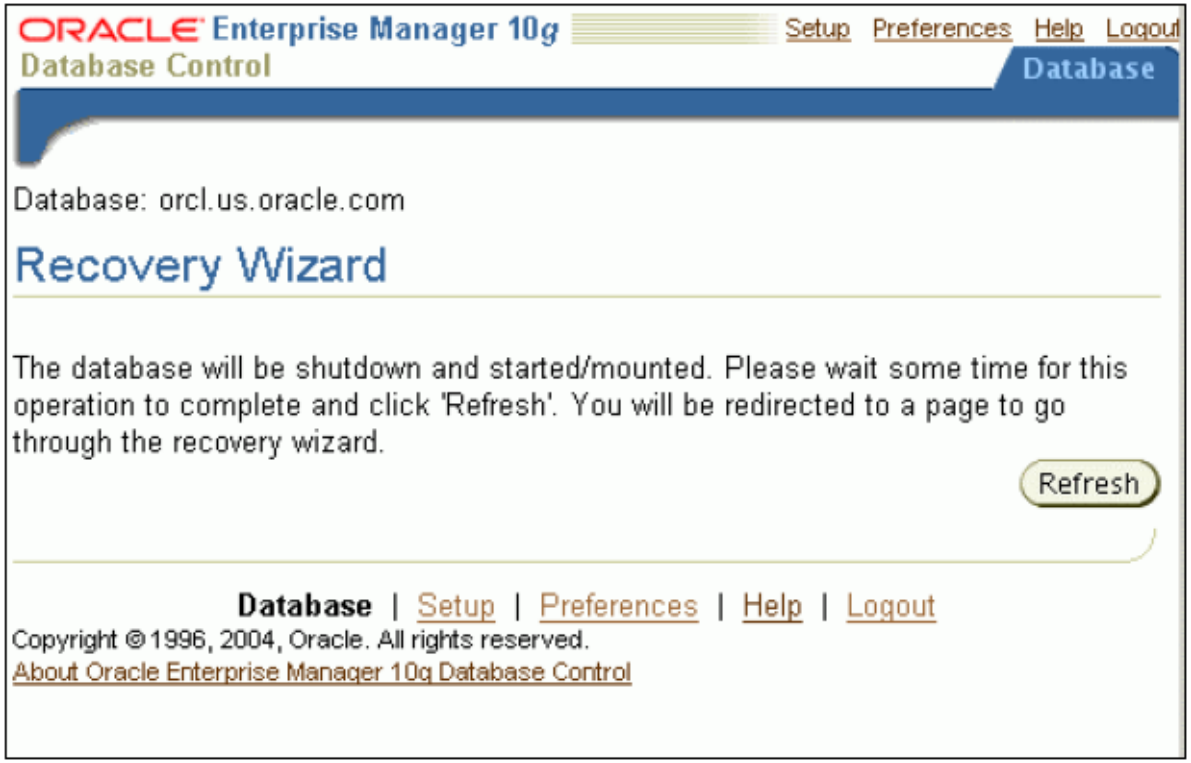
Select object and operation type



Use the Enterprise Manager Recovery Wizard

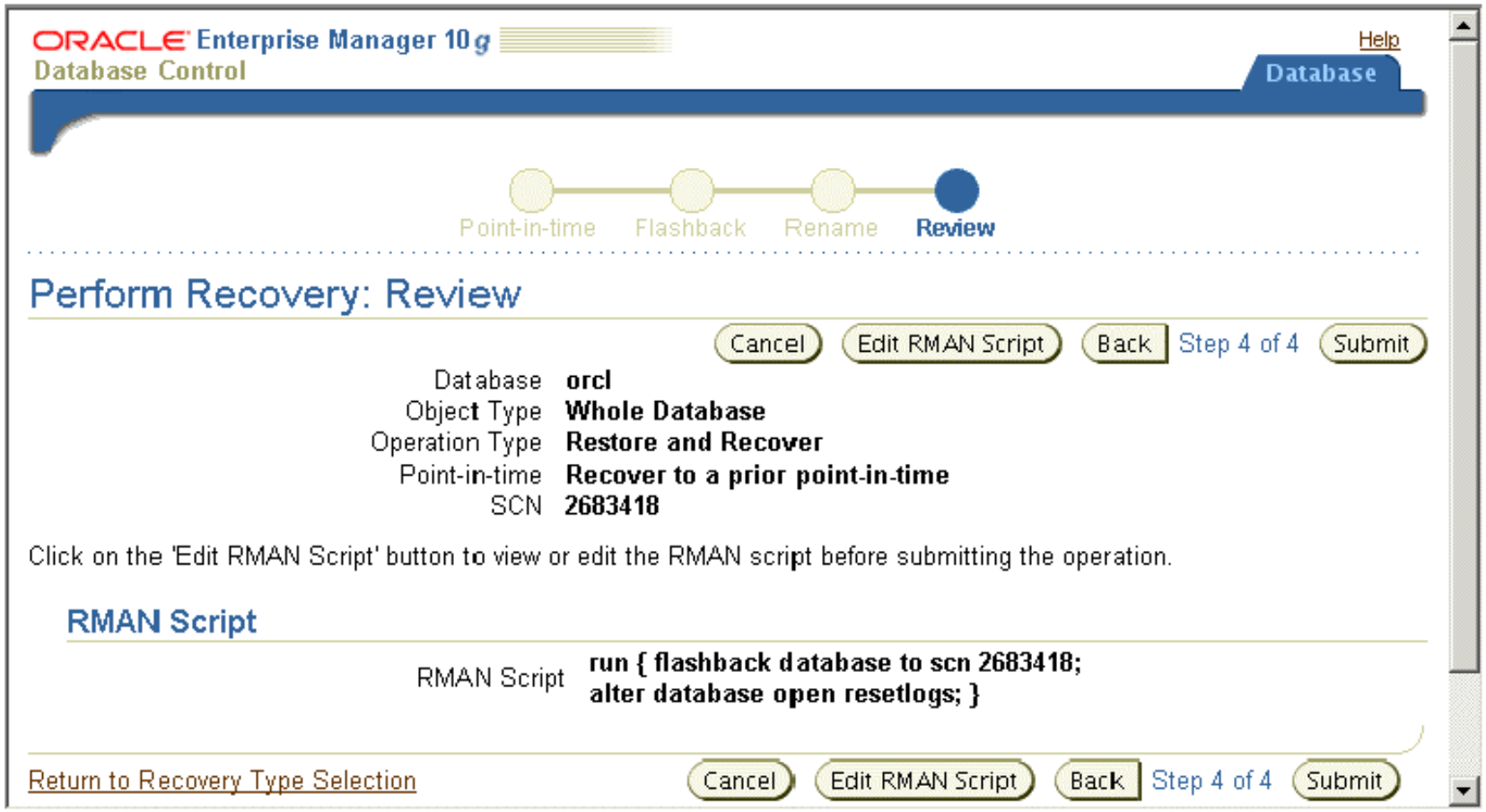
Flashback Database with EM

- Launching Recovery Wizard



Use the Enterprise Manager Recovery Wizard

Flashback Database Using EM



Part 3 Summary

Describe Flashback Database architecture

Configuring Flashback Database

Monitor the Flashback Database

Use the Enterprise Manager Recovery Wizard

Part 3 Stop-and-think

Do you have any questions ?

Recovering from User Errors

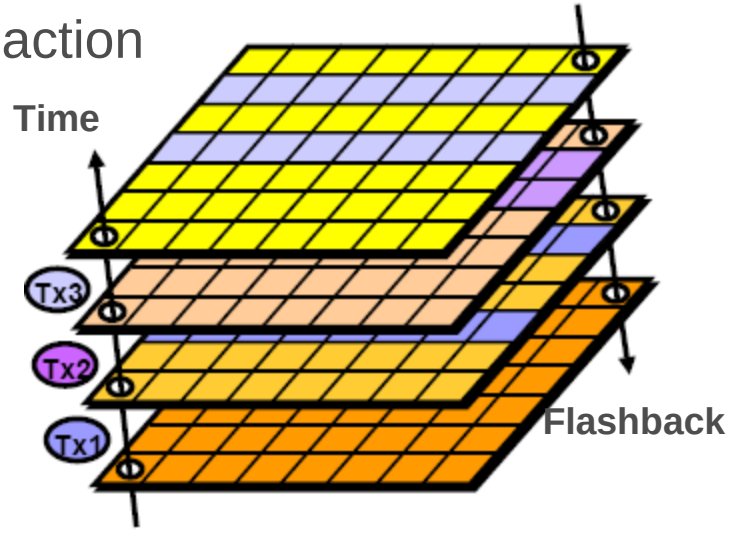
Preview

- Perform Flashback operations
- Manage the recycle bin
- Using Flashback Versions Query
- Flashback Tables
- Perform transaction level recovery

Perform Flashback operations

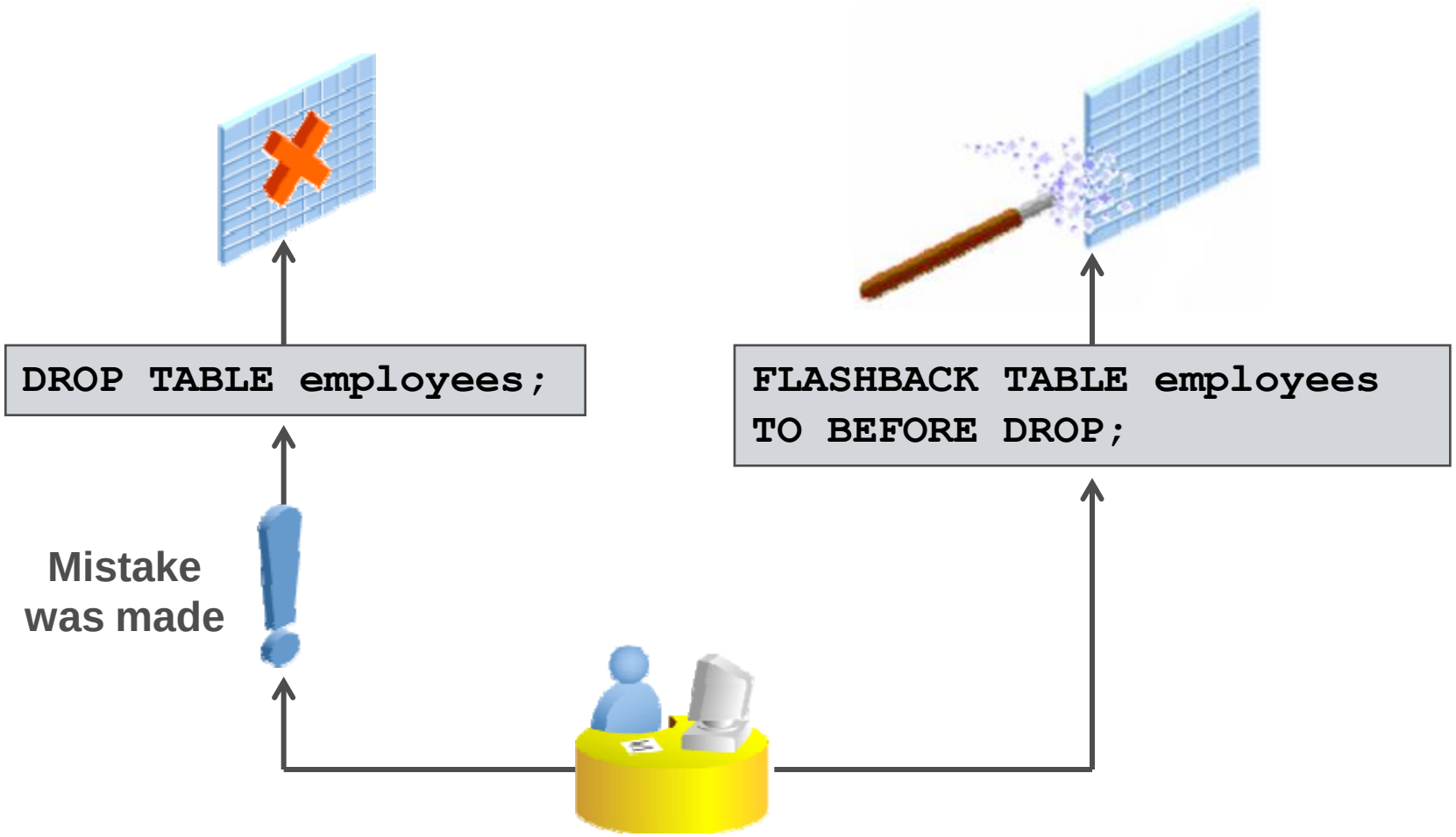
Flashback Time Navigation

- Flashback Query
  - Query all data at a specified point in time•
- Flashback Versions Query
  - See all versions of a row between two times
  - See the transactions that changed the row
- Flashback Transaction Query
  - See all changes made by a transaction



Perform Flashback operations

Flashback Drop Overview



Perform Flashback operations

SCN and Time Mapping Enhancements

- The mapping granularity is three seconds.
- The mapping is retained for

  Max(five days, **UNDO_RETENTION**)

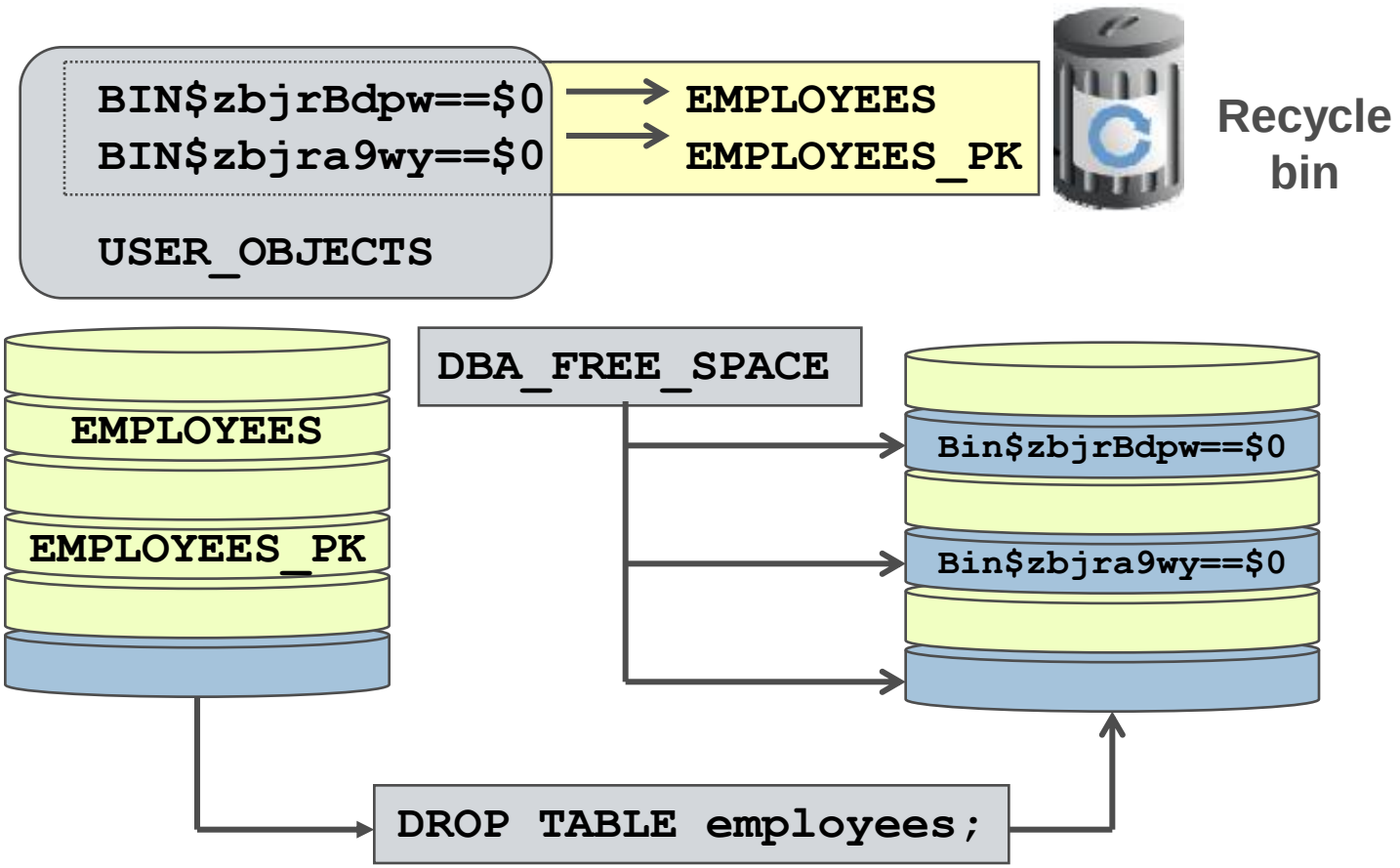
- Access the mapping by using the following SQL functions:
  - **SCN_TO_TIMESTAMP**
  - **TIMESTAMP_TO_SCN**

```
SELECT current_scn, SCN_TO_TIMESTAMP(current_scn)
FROM   v$database;
```

CURRENT_SCN	SCN_TO_TIMESTAMP(CURRENT_SCN)
3274940	23-SEP-03 02.56.03.000000000 AM

Manage the recycle bin

Recycle Bin



Manage the recycle bin

Querying the Recycle Bin

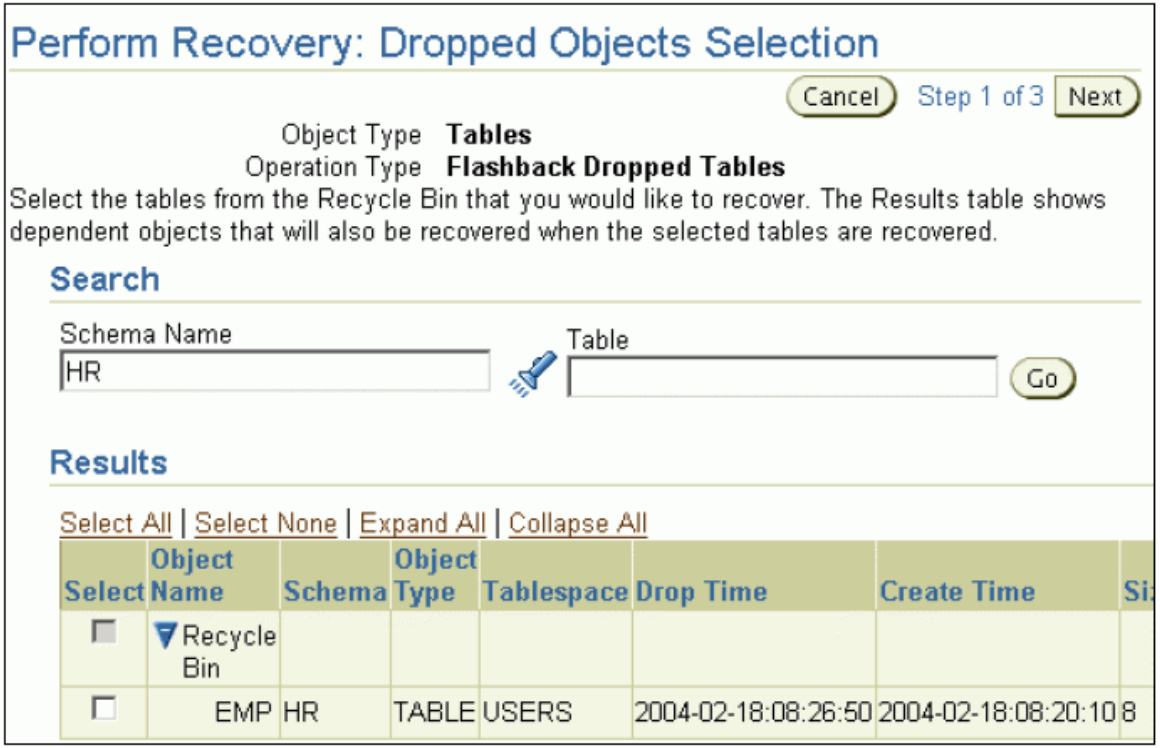
```
SELECT owner, original_name, object_name,
       type, ts_name, droptime, related, space
FROM   dba_recyclebin
WHERE  can_undrop = 'YES';
```

```
SELECT original_name, object_name,
       type, ts_name, droptime, related, space
FROM   user_recyclebin
WHERE  can_undrop = 'YES';
```

```
SQL> SHOW RECYCLEBIN
```

Manage the recycle bin

Flashback Dropped Tables Using EM



Manage the recycle bin

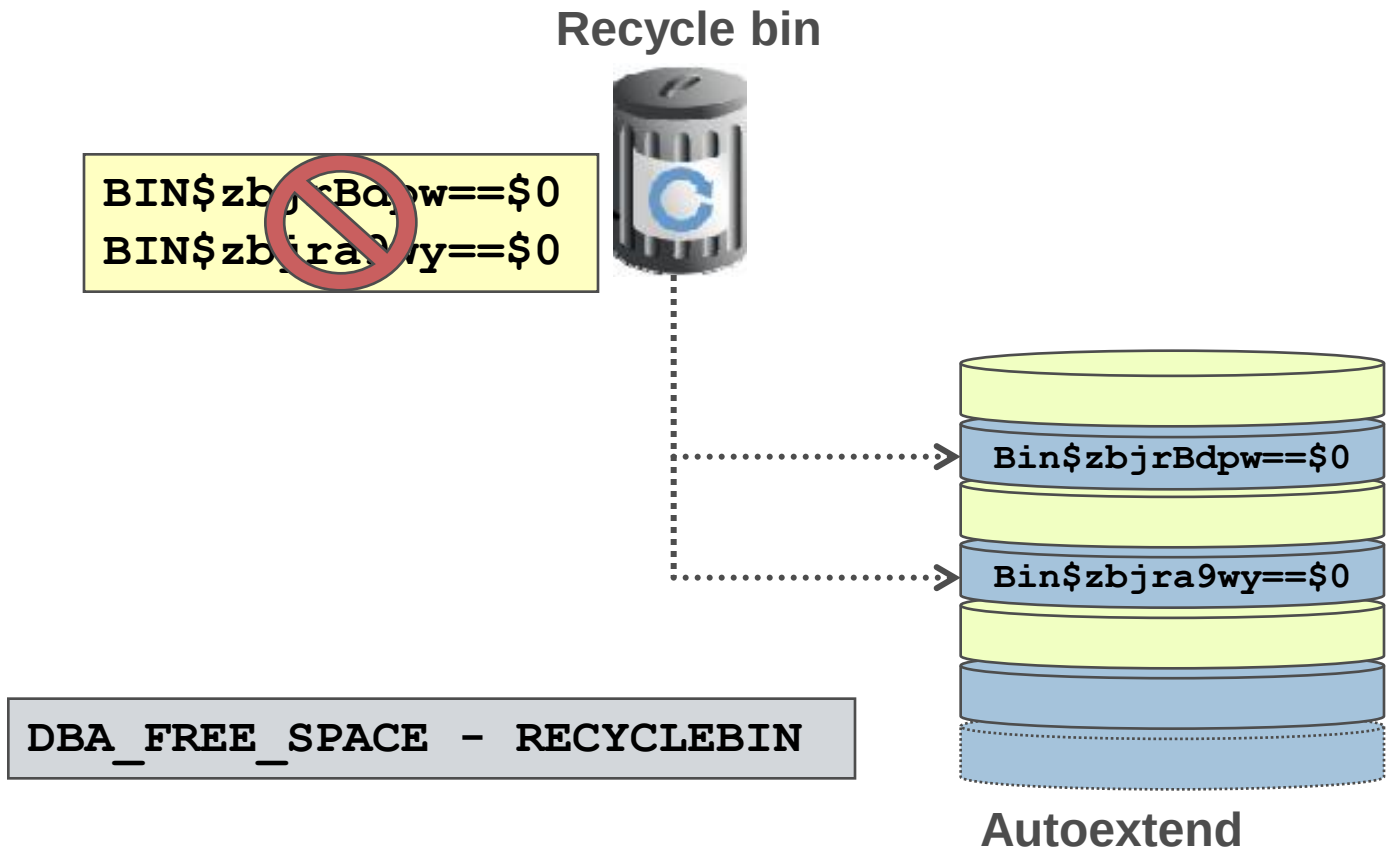
Restoring Objects from the Recycle Bin

- Use the **`FLASHBACK TABLE`** … command to restore dropped tables and dependent objects.
- If multiple recycle bin entries have the same original name:
  - Use unique system-generated names to restore a particular version.
  - When using original names, restored table is LIFO.
- Rename the original name if that name is currently used.

```
FLASHBACK TABLE <table_name>
TO BEFORE DROP [RENAME TO <new_name>]
```

Manage the recycle bin

Recycle Bin Automatic Space Reclamation



`DBA_FREE_SPACE - RECYCLEBIN`

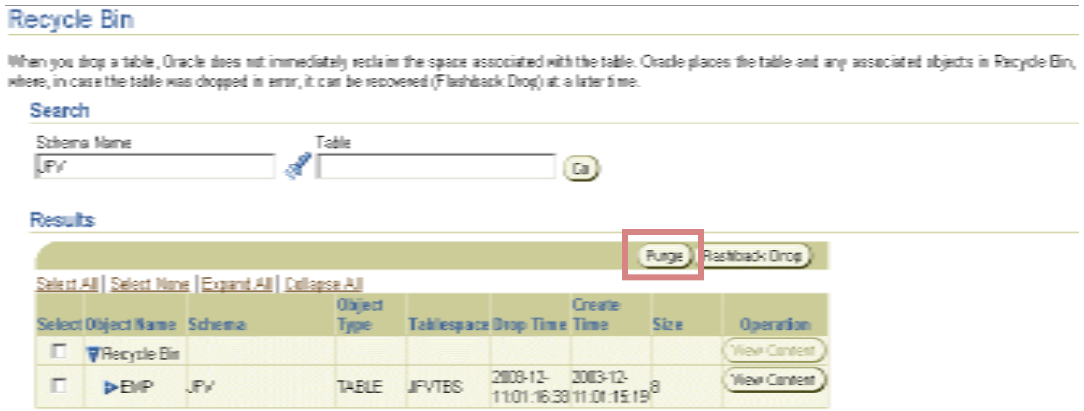
Manage the recycle bin

Recycle Bin Manual Space Reclamation

```
PURGE {TABLE <table_name>|INDEX <index_name>}
```

```
PURGE TABLESPACE <ts_name> [USER <user_name>]
```

```
PURGE [USER_|DBA_]RECYCLEBIN
```



Manage the recycle bin

Bypassing the Recycle Bin

```
DROP TABLE <table_name> [PURGE];
```

```
DROP TABLESPACE <ts_name>
[INCLUDING CONTENTS] ;
```

```
DROP USER <user_name> [CASCADE] ;
```

Manage the recycle bin

Querying Dropped Tables

USER_OBJECTS	
DROPPED	TABLE_NAME
NO	SALES
YES	BIN$zbjrBdpw==$0 → EMPLOYEES
YES	BIN$zbjra9wy==$0 → EMPLOYEES_PK
NO	SALES_PK

DROPPED	INDEX_NAME

USER_INDEXES

Recycle bin

```
SELECT ...
FROM BIN$zbjrBdpw==$0 [AS OF ...]
WHERE ...
```

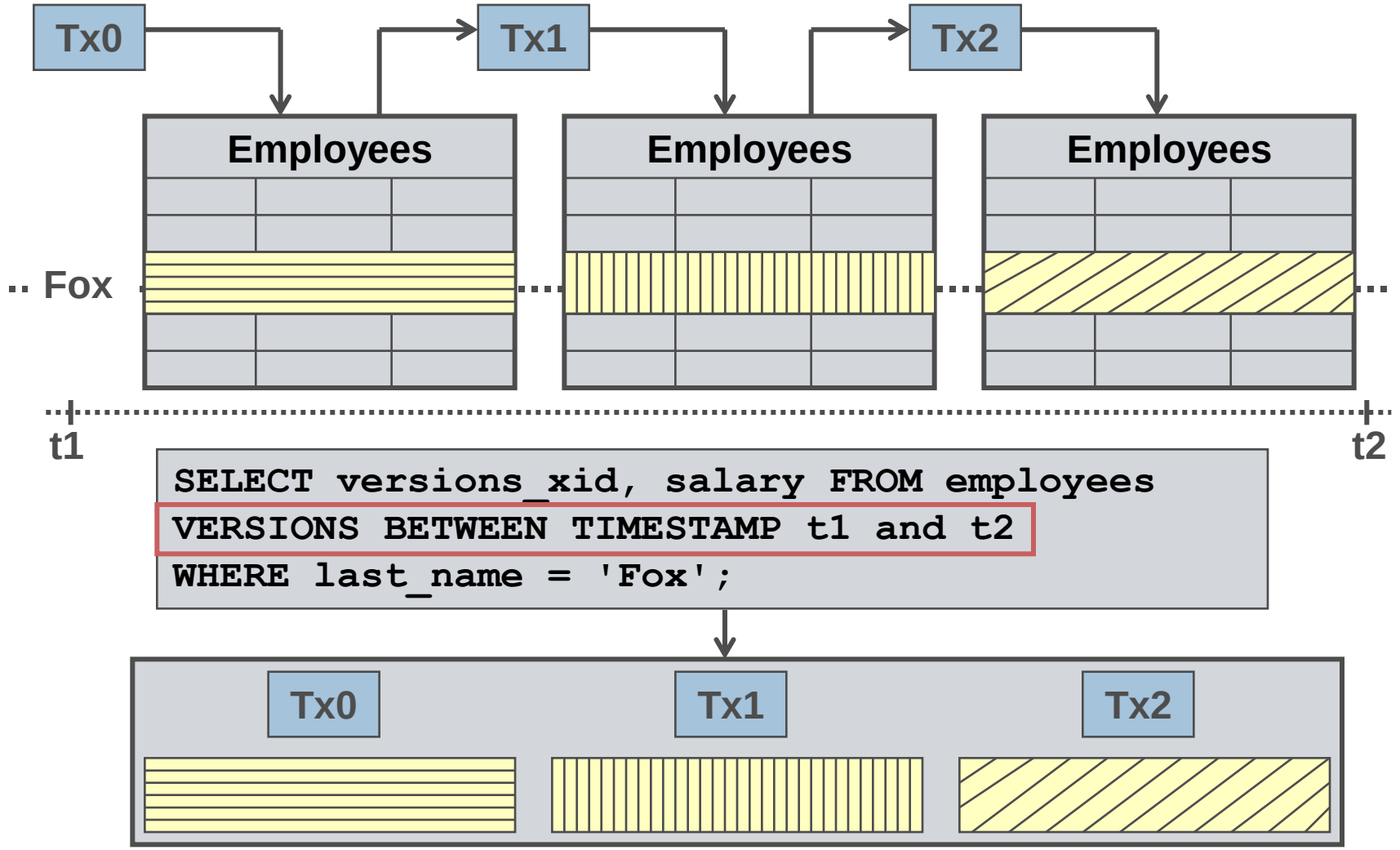
Manage the recycle bin

Flashback Drop Considerations

- Protected tables:
  - Are non-**SYSTEM** tablespace tables
  - Are stored in locally managed tablespaces
  - Do not use fine-grained auditing or virtual private database
- The following dependencies are not protected:
  - Bitmap-join indexes
  - Materialized view logs
  - Referential integrity constraints
  - Indexes dropped before tables
- Purged tables cannot be flashed back

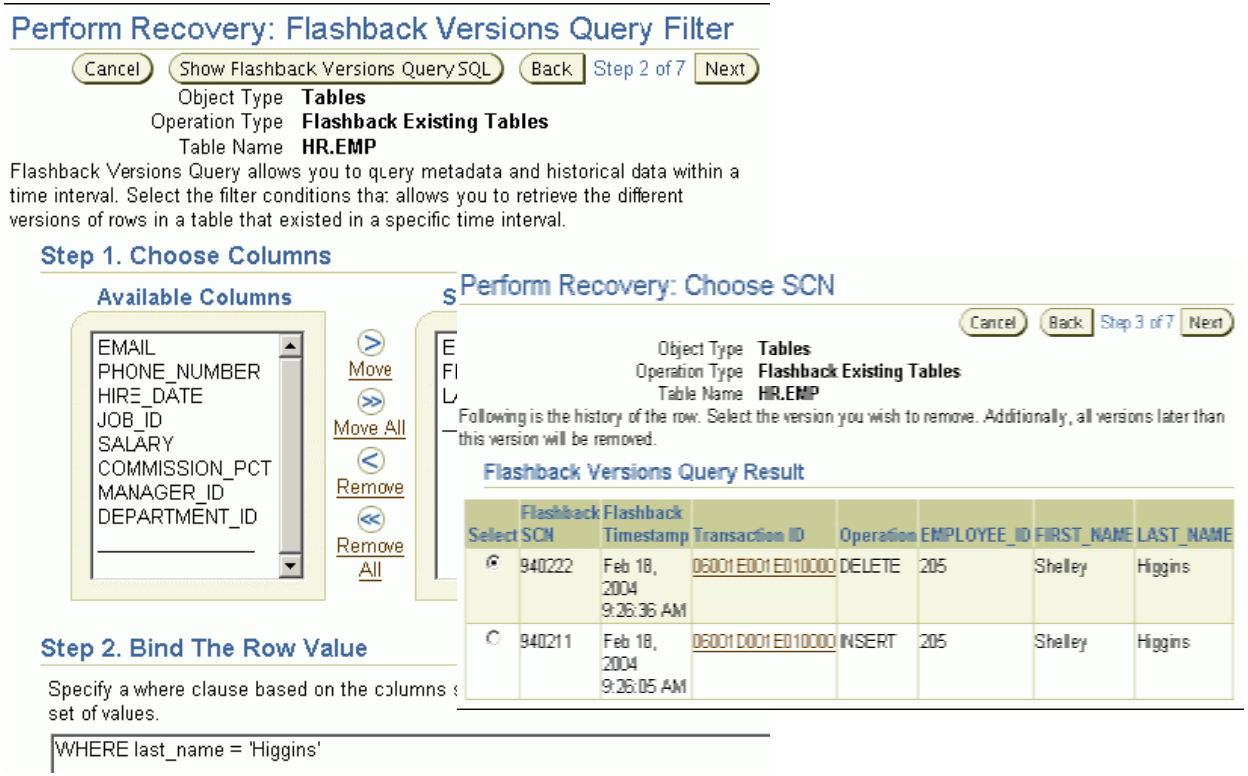
Using Flashback Versions Query

Flashback Versions Query Overview



Using Flashback Versions Query

Flashback Versions Query Using EM



Using Flashback Versions Query

Flashback Versions Query Syntax

```
SELECT [Pseudocolumns]…
FROM …
       VERSIONS BETWEEN
       {SCN|TIMESTAMP{expr|MINVALUE}AND
                     {expr|MAXVALUE}}
        [AS OF{SCN|TIMESTAMP expr}]
WHERE [Pseudocolumns…]…
```

Pseudocolumn	Description
VERSIONS_STARTTIME VERSIONS_STARTSCN	Version validity range lower bound
VERSIONS_ENDTIME VERSIONS_ENDSCN	Version validity range upper bound
VERSIONS_XID	Transaction that created the version
VERSIONS_OPERATION	Operation that produced the version

Using Flashback Versions Query

Flashback Versions Query Example

```
SELECT versions_xid AS XID,
       versions_startscn AS START_SCN,
       versions_endscn AS END_SCN,
       versions_operation AS OPERATION,
       first_name
FROM   employees
VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
AS OF SCN 5525300
WHERE  employee_id = 111;
```

XID	START_SCN	END_SCN	O	FIRST_NAME
8C0025003A000000	5525293		I	Tom
8C0024003A000000	5525291		D	Mike
8C0022003A000000	5525277	5525291	I	Mike

Using Flashback Versions Query

Flashback Versions Query Considerations

- The **`VERSIONS`** clause cannot be used to query:
  - External tables
  - Temporary tables
  - Fixed tables
  - Views
- The **`VERSIONS`** clause cannot span DDL commands.
- Segment shrink operations are filtered out.

Using Flashback Versions Query

Guaranteed Undo Retention

```
SQL> CREATE UNDO TABLESPACE undotbs1
   2 DATAFILE 'undotbs01.dbf'
   3 SIZE 100M AUTOEXTEND ON
   4 RETENTION GUARANTEE ;
```

```
SQL> SELECT tablespace_name, RETENTION
   2 FROM dba_tablespaces;
```

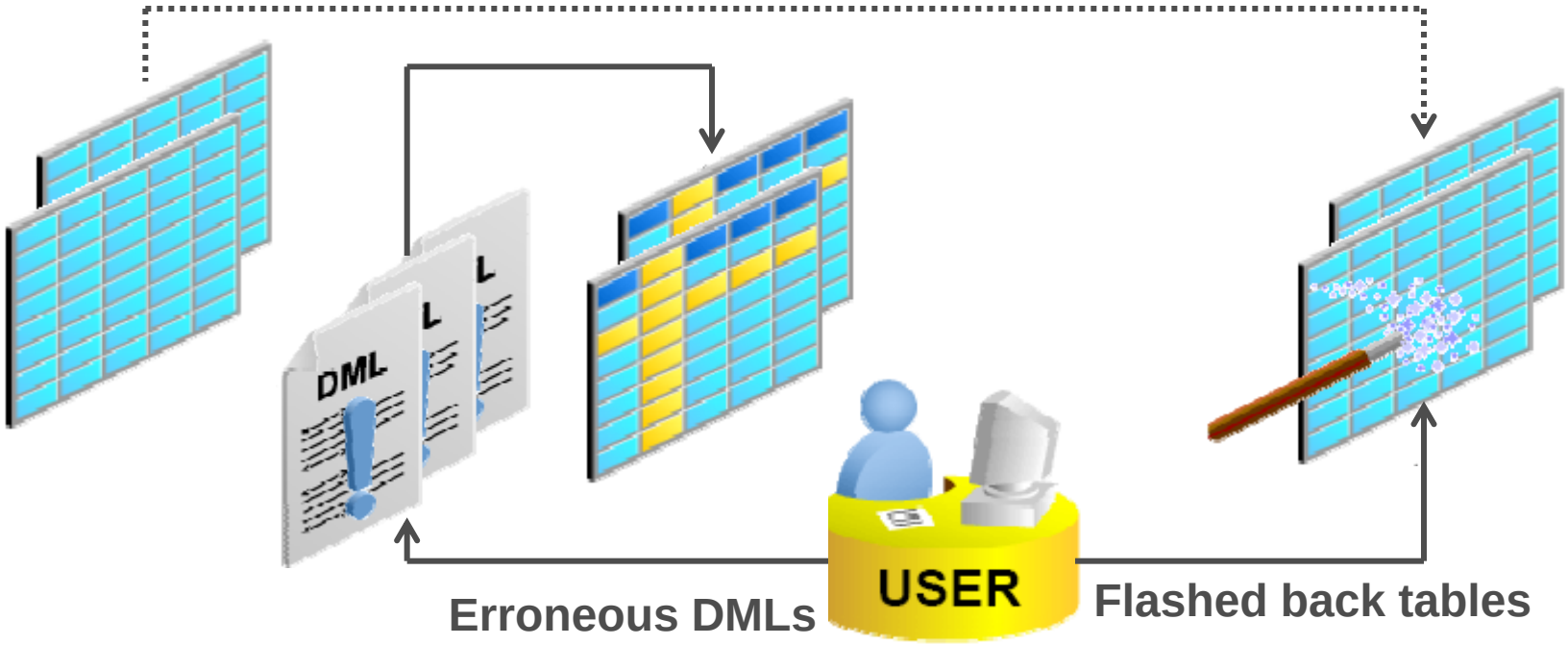
TABLESPACE_NAME	RETENTION
UNDOTBS1	GUARANTEE

```
SQL> ALTER TABLESPACE undotbs1
   2 RETENTION NOGUARANTEE;
```

Flashback Tables

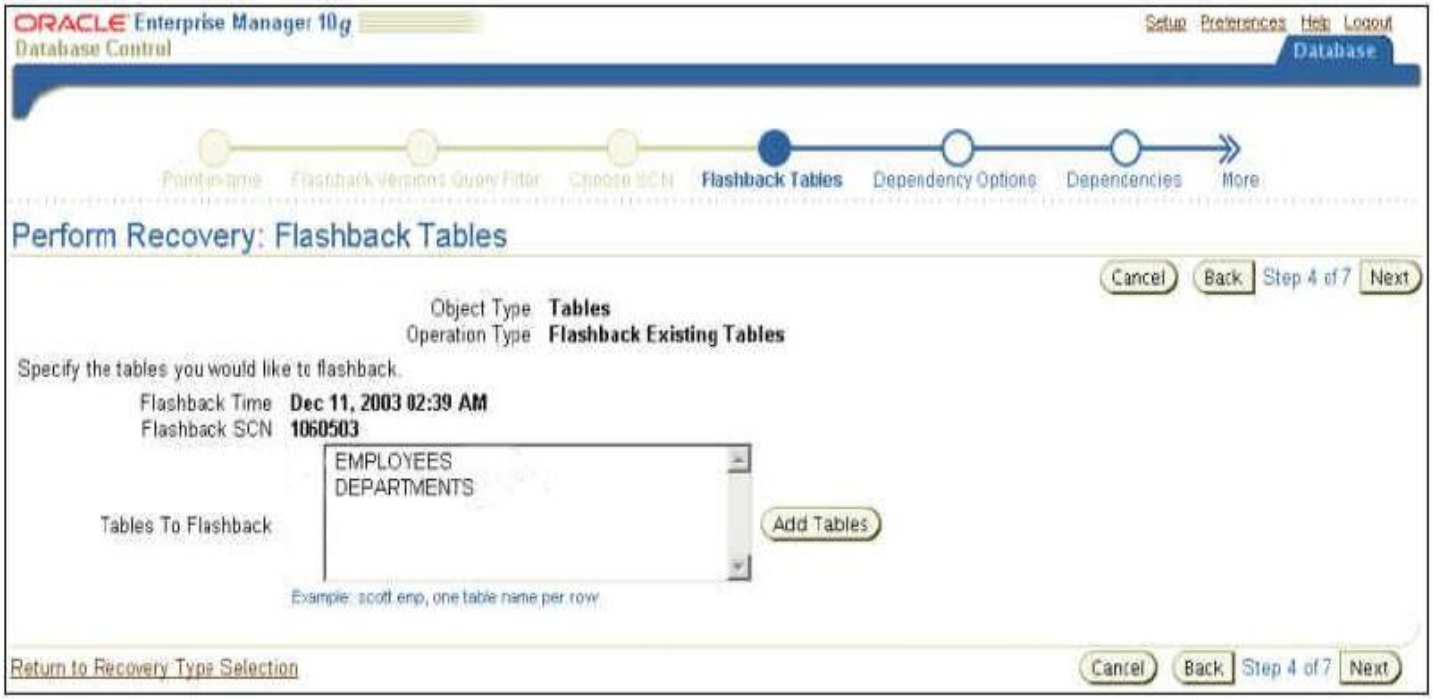
Flashback Table Overview

- Recover tables to a specific point in time
- Flashback Table is an in-place operation
- Database stays online



Flashback Tables

Using EM to Flashback Tables



Flashback Tables

Flashback Table Example

```
ALTER TABLE employees ENABLE ROW MOVEMENT;
```

```
FLASHBACK TABLE employees
TO TIMESTAMP (SYSDATE-1);
```

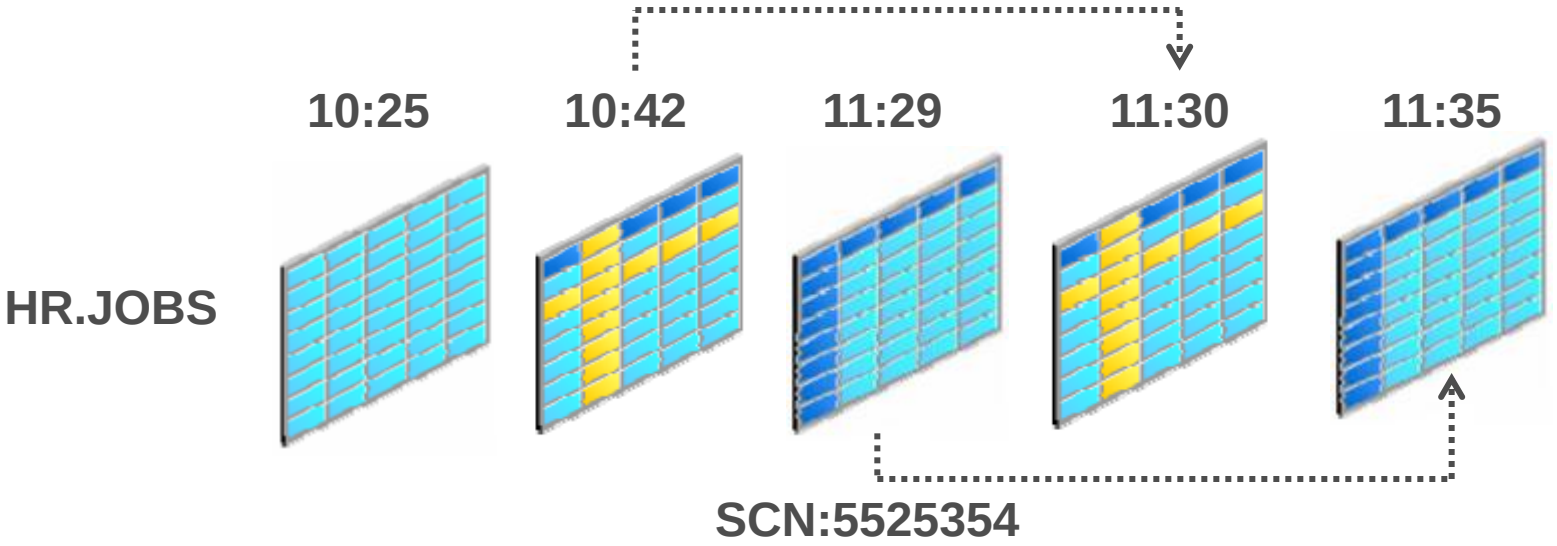
```
ALTER TABLE employees ENABLE ROW MOVEMENT;
```

```
ALTER TABLE departments ENABLE ROW MOVEMENT;
```

```
FLASHBACK TABLE employees, departments
TO SCN 5525293 ENABLE TRIGGERS;
```

Flashback Tables

Rolling Back a Flashback Table Operation



11:30

```
FLASHBACK TABLE jobs
TO TIMESTAMP to_timestamp('10:42','hh24:mi');
```

11:35

```
FLASHBACK TABLE jobs
TO SCN 5525354;
```

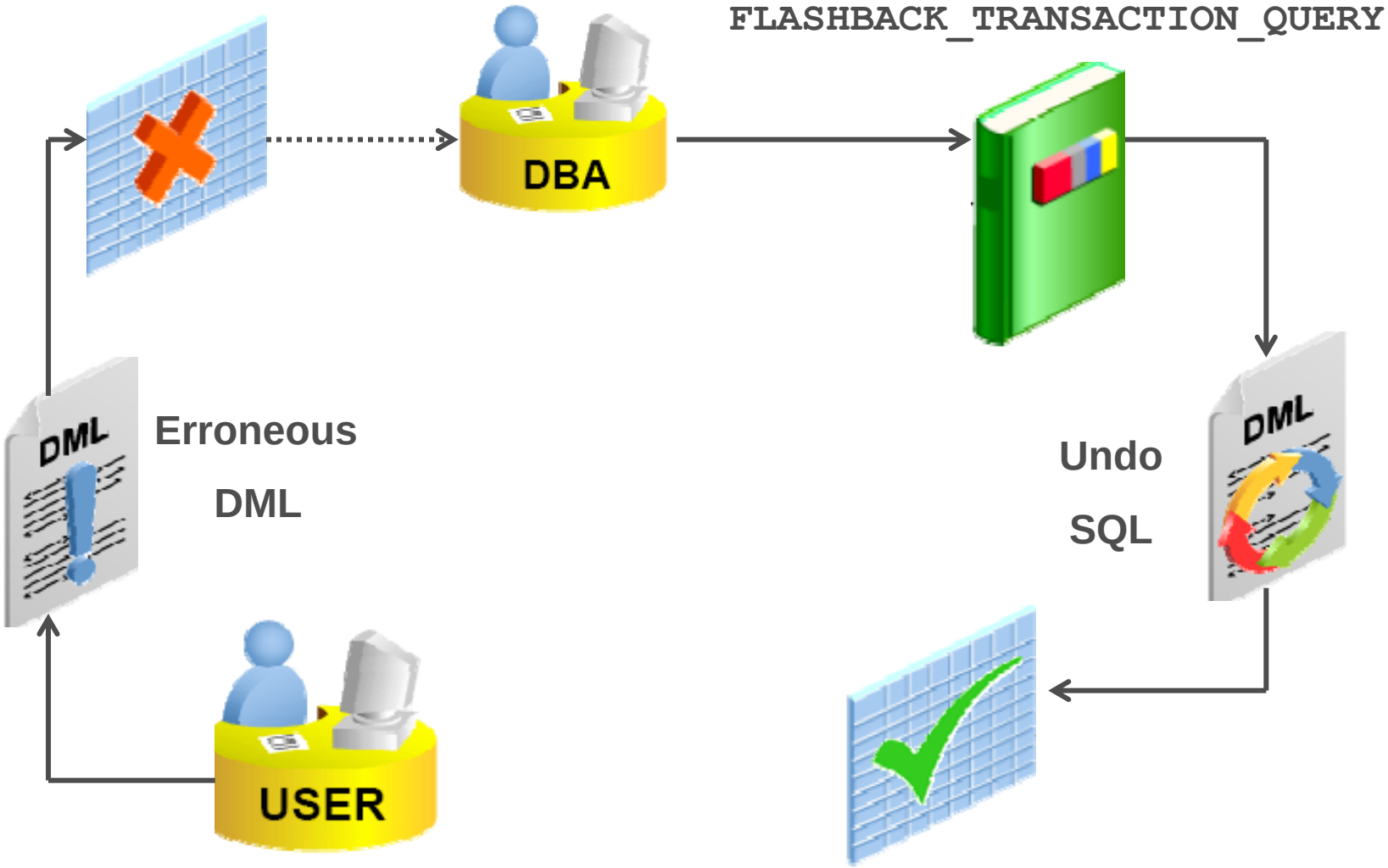
Flashback Tables

Flashback Table Considerations

- The **`FLASHBACK TABLE`** command executes as a single transaction, acquiring exclusive DML locks.
- Statistics are not flashed back.
- Current indexes and dependent objects are maintained.
- Flashback Table operations:
  - Cannot be performed on system tables
  - Cannot span DDL operations
  - Are written to the alert log file

Perform transaction level recovery

Flashback Transaction Query Overview



Perform transaction level recovery

Querying FLASHBACK_TRANSACTION_QUERY

```
SELECT operation, undo_sql, table_name
FROM   FLASHBACK_TRANSACTION_QUERY;
```

```
SELECT operation, undo_sql, table_name
FROM   FLASHBACK_TRANSACTION_QUERY
WHERE  xid = HEXTORAW('8C0024003A000000')
ORDER BY undo_change#;
```

```
SELECT operation, undo_sql, table_name
FROM   FLASHBACK_TRANSACTION_QUERY
WHERE  start_timestamp >= TO_TIMESTAMP
 ('2003-10-21 11:00:00','YYYY-MM-DD HH:MI:SS')
 AND commit_timestamp <= TO_TIMESTAMP
 ('2003-10-21 11:30:00','YYYY-MM-DD HH:MI:SS');
```

Perform transaction level recovery

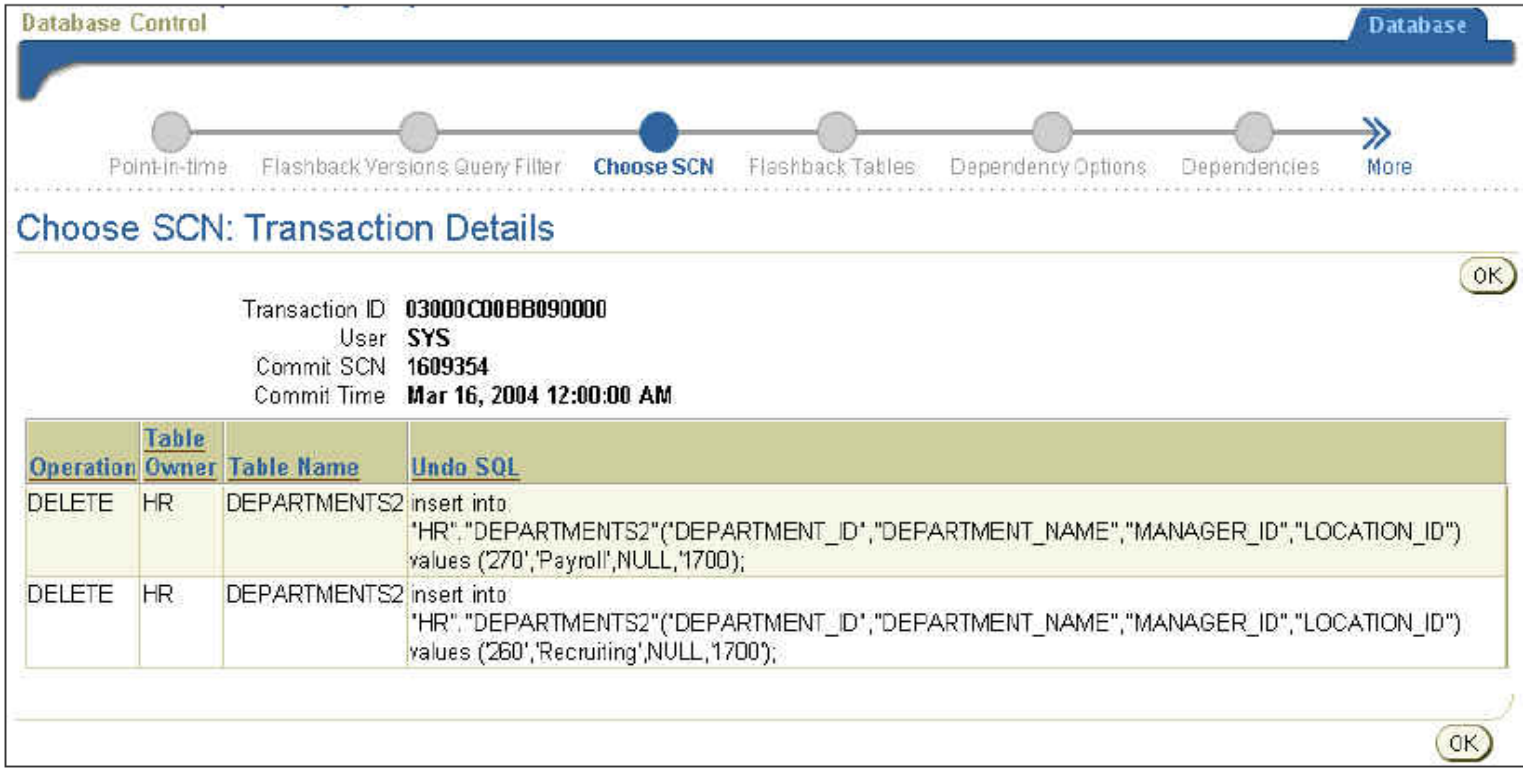
Using Flashback Versions Query and Flashback Transaction Query

```
SELECT versions_xid ,first_name
FROM hr.employees
VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE employee_id = 111;
```

```
SELECT operation, undo_sql
FROM FLASHBACK_TRANSACTION_QUERY
WHERE xid= HEXTORAW('8C0024003A000000');
```

Perform transaction level recovery

Flashback Transaction Query Using EM



Perform transaction level recovery

Flashback Transaction Query Considerations

- DDLs are seen as dictionary updates.
- Dropped objects appear as object numbers.
- Dropped users appear as user identifiers.
- Minimal supplemental logging may be needed:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

Part 4 Summary

- Perform Flashback operations
- Manage the recycle bin
- Flashback Tables
- Using Flashback Versions Query
- Perform transaction level recovery

Part 4 Stop-and-think

Do you have any questions ?

THE INTERNATIONAL INSTITUTE
SUPINFO
OF INFORMATION TECHNOLOGY

Congratulations

You have successfully completed
the SUPINFO course n°33

Oracle Technologies
Managing Data Recovery

The end


THE INTERNATIONAL INSTITUTE
SUPINFO
OF INFORMATION TECHNOLOGY