



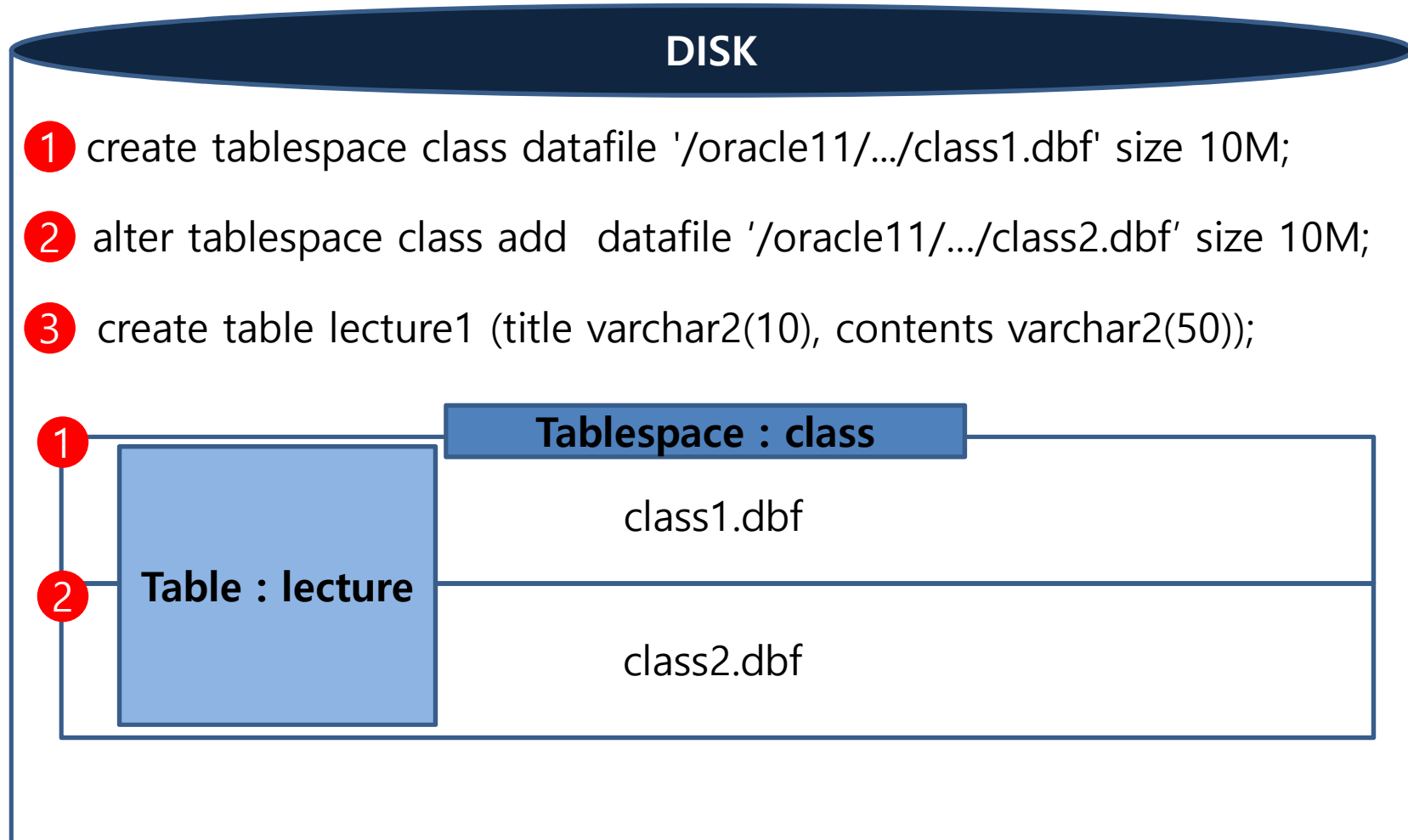
Part 1 .Administration

1. Oracle Architecture
2. Oracle SGA (System Global Area)
3. Oracle Background Processes
4. Oracle Startup & Shutdown
5. Control files
6. Redo log files
- 7. Tablespace & Data files** ←
8. User Managements

7. Tablespace & Datafiles

- 오라클은 데이터를 저장하고 작업할 때 메모리에 논리적으로는 Tablespace라는 공간을 만들어서 작업하며, 물리적으로는 Data File을 만들어서 저장
- 논리적인 tablespace라는 공간에 여러 개의 table과 index가 생성된다

7. Tablespace & Datafiles



7. Tablespace & Datafiles



2. Tablespace의 종류 및 특징

1) SYSTEM Tablespace

- Data Dictionary들이 저장되어있음
- SYS 계정의 소유 Tablespace이지만 SYS 사용자도 이 테이블의 내용을 변경할 수 없음
- Data Dictionary 테이블 개수 조회
SQL> select count(*) from dictionary;

7. Tablespace & Datafiles



- Data dictionary : 오라클 서버의 모든 정보를 저장하고 있는 아주 중요한 테이블이나 뷰
- Data dictionary에 있는 정보
 1. 데이터베이스의 논리적인 구조와 물리적인 구조 정보
 2. 객체의 정의와 공간 활용 정보
 3. 제약조건에 관련된 정보
 4. 사용자에게 관련된 정보
 5. Role, Privilege 등에 관련된 정보
 6. 감사 및 보안에 관련된 정보

7. Tablespace & Datafiles

- Data Dictionary는 Base Table과 Data Dictionary View로 나뉨
 - Base Table : 데이터베이스를 생성할 때 생성되는 테이블로 Dictionary의 원본 데이터가 존재
 - Data Dictionary View : dbca로 database 생성 시 자동생성 되지만, create databas로 수동 생성 경우 catalog.sql을 수행해야 생성됨
 - Base Table의 내용을 직접 보지 못하도록 Data Dictionary View를 제공하는 것
 - Data Dictionary View는 다시 Static Dictionary와 Dynamic Dictionary로 나뉨

7. Tablespace & Datafiles



- Static Dictionary
 - 수동으로 업데이트를 해야 정보가 갱신
 - 인스턴스가 OPEN 되었을 경우 조회가능
 - USER_XXX, ALL_XXX, DBA_XXX 뷰들
- Dynamic Performance View
 - 실시간으로 변경되는 내용을 볼 수 있는 뷰
 - 사용자가 조회할 경우 그 시점의 Control File 이나 메모리에서 데이터를 가져와 보여준다
 - v\$로 시작되며, Database가 Nomount 상태일 때 부터 조회가능

7. Tablespace & Datafiles

2) SYSAUX Tablespace

- 주로 오라클 서버의 성능 튜닝을 위한 데이터들이 저장되어 있음
- 9i 버전까지는 튜닝관련 Dictionary들이 SYSTEM Tablespace에 존재
- 10g버전부터 성능 튜닝과 관련된 Dictionary들이 이곳에 별도로 저장
- 10g에 새로 등장한 자동튜닝 기능들 AWR, ADDM, ASH 등이 이곳의 정보를 사용

7. Tablespace & Datafiles

3) 일반 Tablespace

- 일반적으로 많이 사용되는 Tablespace
- 관리자가 필요에 의해 만드는 Tablespace

7. Tablespace & Datafiles

실습1. 일반 Tablespace 생성 및 조회하기

Tablespace 생성하기

```
SQL> create tablespace haksa  
2 datafile '/home/oracle/oradata/testdb/haksa01.dbf' size 1M;
```

Tablespace 조회하기

```
SQL> select tablespace_name, status, contents, extent_management,  
2 segment_space_management  
3 from dba_tablespaces;
```

```
SQL> select tablespace_name, bytes/1024/1024 MB , file_name  
2 from dba_data_files ;
```

7. Tablespace & Datafiles

실습2. 각 Data File의 실제 사용량 확인하는 방법

```
SQL> select distinct d.file_id          file#,
2  d.tablespace_name                    ts_name,
3  d.bytes /1024 / 1024                 MB,
4  d.bytes / 8192                      total_blocks,
5  sum(e.blocks)                       used_blocks,
6  to_char( nvl( round( sum(e.blocks)/(d.bytes/8192), 4),0) *100,'09.99') ||
   ' %' pct_used
7  from  dba_extents e, dba_data_files d
8  where d.file_id = e.file_id(+)
9  group by d.file_id , d.tablespace_name , d.bytes
10 order by 1,2 ;
```

7. Tablespace & Datafiles

실습3. Tablespace 용량 관리 하기

Tablespace Full 발생 유도

```
SQL> create table iphak (stduno number) tablespace haksa;
```

```
SQL> begin
  2  for i in 1..50000 loop
  3  insert into iphak values(i);
  4  end loop
  5  ;
  6  commit ;
  7  end;
  8  /
```

```
SQL> /
```

ORA-01653 : unable to extend error 발생

7. Tablespace & Datafiles

실습3. Tablespace 용량 관리 하기

조치1. 데이터 파일 추가

```
SQL> alter tablespace haksa  
2 add datafile '/home/oracle/oradata/testdb/haksa02.dbf' size 20M;
```

조치2. Data File 크기 수동 증가시키기

```
SQL> alter database datafile '/home/oracle/oradata/testdb/haksa01.dbf'  
2 resize 20M;
```

7. Tablespace & Datafiles

실습3. Tablespace 용량 관리 하기

조치3. Data File 크기 자동 증가시키기

- Autoextend 사용 (자동 증가)

```
SQL> alter database datafile '/home/oracle/oradata/testdb/haksa01.dbf'  
      autoextend on ;
```

- 32 bit일 경우 최대 16GB, 64 bit일 경우 최대 32GB까지 확장됨
- autoextend 확인

```
SQL> select tablespace_name, byte/1024/1024 MB, file_name,  
           autoextensible  
       from dba_data_files ;
```

7. Tablespace & Datafiles

실습3. Tablespace 용량 관리 하기

Autoextend TEST

```
SQL> begin
  2  for i in 1..50000 loop
  3  insert into iphak values(i);
  4  end loop
  5  ;
  6  commit ;
  7  end;
  8  /
```

```
SQL> /
```

SQL> / -> SIZE 조회하면 Data_file이 자동으로 확장되어 있음

7. Tablespace & Datafiles

- Tablespace Offline
 - 더 이상 해당 Tablespace에 접근하지 않겠다는 의미로 읽기도 쓰기도 되지 않는 상태
 - Tablespace의 위치 이동, 장애 복구 시 주로 사용
 - Normal, Temporary, Immediate Mode 존재

7. Tablespace & Datafiles

1) Normal Mode

- 가장 일반적인 offline 방법
- Tablespace가 정상일 경우만 가능
- `alter tablespace tablespace_name offline;`

2) Temporary Mode

- offline 시키고자 하는 Tablespace의 Data file에 하나라도 이상이 생기게 될 경우 수행
- `alter tablespace tablespace_name offline temporary ;`

7. Tablespace & Datafiles

3) Immediate Mode

- Data file에 장애가 나서 데이터를 내려쓰지 못하는 상황에서 offline을 해야 하는 경우
- 반드시 archive log mode일 경우에만 사용
- 나중에 online시 복구하라고 메시지 나옴
- `alter tablespace haksu offline immediate ;`

7. Tablespace & Datafiles

- Data File Offline
 - Data File 단위로도 offline이 가능
 - archive mode 경우
: alter database datafile '.....' offline;
 - No archive mode 일 경우에는 offline drop
: alter database datafile '.....' offline drop;
 - Data file offline은 online 전에 recover 필요

7. Tablespace & Datafiles

실습4. Tablespace Offline

1. Normal Mode Offline / Online

```
SQL> alter tablespace haksa offline ;
```

```
SQL> alter tablespace haksa online ;
```

Offline 후 online 하면 데이터파일의 Checkpoint 정보 다름.
Alter system checkpoint ; 로 동기화 작업 필요함

2. Datafile 단위 Offline

```
SQL> alter database datafile '/home/oracle/oradata/testdb/haksa01.dbf'  
offline drop;
```

3. Temporary Mode Offline

```
SQL> alter tablespace haksa offline ; -- 실패(1개의 datafile offline이므로)  
SQL> alter tablespace haksa offline temporary ;
```

7. Tablespace & Datafiles

실습4. Tablespace Offline

4. recover가 필요한 datafile online

```
SQL> alter tablespace haksa online; -- recover 필요하다고 error
```

```
SQL> recover tablespace haksa;
```

```
SQL> alter tablespace haksa online;
```

5. CHECKPOINT 확인 및 강제 동기화 필요

7. Tablespace & Datafiles

- Data File Rename
 - 특정 디스크에 있는 데이터 파일들의 용량이 점점 증가하여 다른 더 큰 용량의 디스크를 설치 한 후 Data file을 이동 시 주로 사용
 - Datafile이 사용 중일 때는 절대로 이동시키거나 복사하면 안된다
 - Tablespace를 Offline 하거나, Shutdown 한 후 작업해야 함
 - 일반 Tablespace는 Offline후 작업
 - Offline이 불가능한 SYSTEM, UNDO, Default Temp Tablespace는 Shutdown 후 작업 가능

7. Tablespace & Datafiles

- 일반 Tablespace Rename 절차
 1. 해당 Tablespace Offline
 2. Data file 물리적 rename(mv보다 cp로 하는 것이 안전)
 3. Control file 정보 변경
 4. 해당 Tablespace Online

7. Tablespace & Datafiles

- SYSTEM Tablespace Rename 절차
 1. shutdown
 2. Startup mount
 3. Data file 물리적 rename
 4. Control file 정보 변경
 5. Open

7. Tablespace & Datafiles

실습1. 일반 Tablespace rename

haksa Tablespace를 /oracle11 위치에서 home 위치로 변경

1. Tablespace offline

```
SQL> alter tablespace haksa offline ;
```

2. Data file 복사

```
SQL> !cp /oracle11/.../haksa01.dbf /home/.../haksa01.dbf
```

```
SQL> !cp /oracle11/.../haksa02.dbf /home/.../haksa02.dbf
```

3. control file 내용 변경

```
SQL> alter tablespace haksa rename  
      datafile '/oracle11/.../haksa01.dbf' to '/home/.../haksa01.dbf';
```

```
SQL> alter tablespace haksa rename  
      datafile '/oracle11/.../haksa02.dbf' to '/home/.../haksa02.dbf';
```

4. Tablespace online

```
SQL> alter tablespace haksa online ;
```

7. Tablespace & Datafiles

실습2. SYSTEM Tablespace rename

SYSTEM Tablespace를 /oracle11 위치에서 home 위치로 변경

1. shutdown

SQL> shutdown immediate

2. startup mount

SQL> startup mount

3. Data file 복사

SQL> !cp /oracle11/.../system01.dbf /home/.../system01.dbf

4. control file 내용 변경

SQL> alter database rename
file '/oracle11/.../system01.dbf' to '/home/.../system01.dbf';

5. Open

SQL> alter database open ;

7. Tablespace & Datafiles

- Tablespace 삭제
 - drop tablespace tablespace_name ;
 - Tablespace에 Table이 하나라도 있으면 삭제되지
않음
 - drop tablespace tablespace_name including
contents and datafiles ; 로 삭제 가능
 - ASM이 아니라면 물리적으로도 data file 삭제 필
요

7. Tablespace & Datafiles



- Undo Tablespace
 - DML 작업 수행 시 원본 데이터(Undo data)를 저장하는 장소
 - Undo Data를 저장하는 실질적 공간을 Undo Segment라고 하고 이 Segment를 저장하는 Tablespace가 Undo Tablespace
 - 기본적으로 하나의 세션당 하나의 Undo segment 를 할당
 - Undo Tablespace에는 Undo Data 외의 다른 데이터는 저장될 수 없다
 - 사용자가 관여하거나 수정할 수 없고 Oracle Server Process가 직접 관리

7. Tablespace & Datafiles

- Undo Tablespace는 Instance당 여러 개가 존재할 수 있으나, 사용되는 것은 한번에 1개뿐
- 관리방법으로는 자동 모드와 수동 모드가 있으며, Default는 자동모드
- `undo_management = auto | manual` 로 관리모드 변경 가능하지만 DB 재기동 필요
- Segment의 개수를 수동으로 관리하는 수동모드와는 달리 자동모드에서는 자동적으로 Segment 할당을 관리, 새로운 Segment를 생성한다

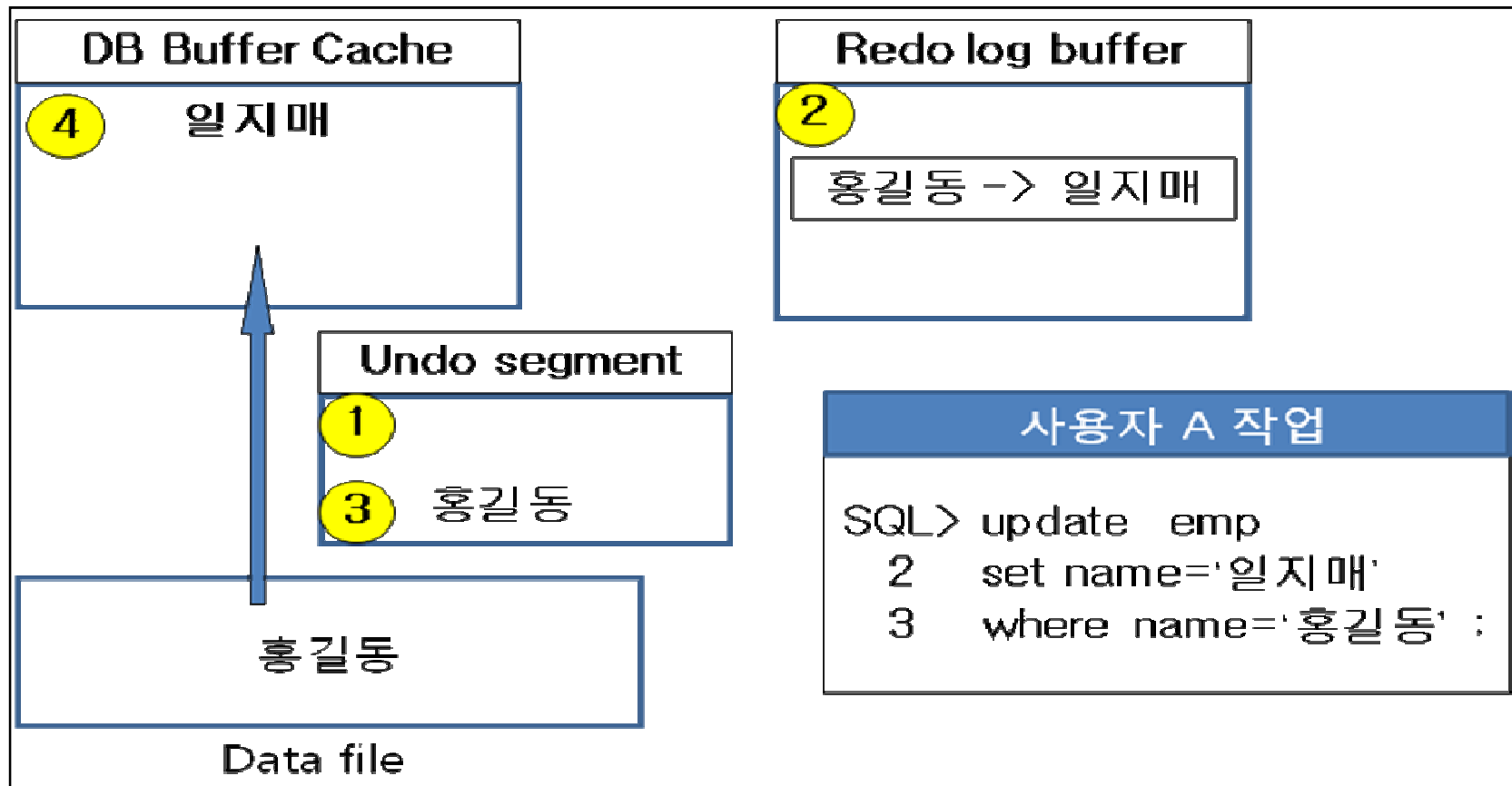
7. Tablespace & Datafiles



- Undo Tablespace 사용 목적
 - 1) Transaction Rollback : 최종 Commit 하지 않고 Rollback 하고자 할 때 Undo 데이터를 이용하여 과거 이미지 전달
 - 2) Transaction Recovery (instance Recovery시 rollback 단계) : Instance Crash 발생 후 Redo를 이용해 Roll forward 단계가 완료되면 최종 Commit 되지 않은 변경사항까지 모두 복구할 때 쓰임
 - 3) Read Consistency : 읽기 일관성을 위해 사용

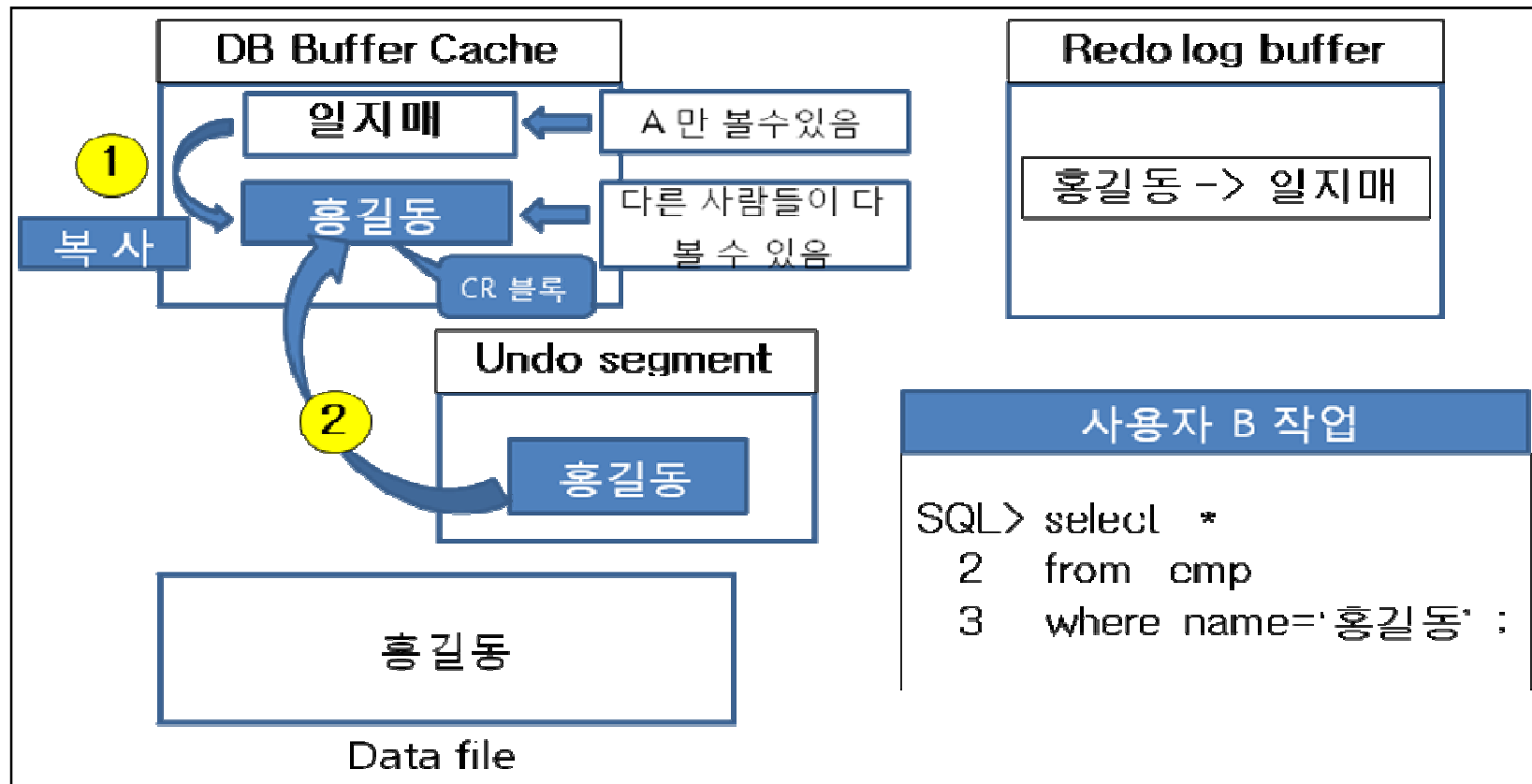
7. Tablespace & Datafiles

Read Consistency -1



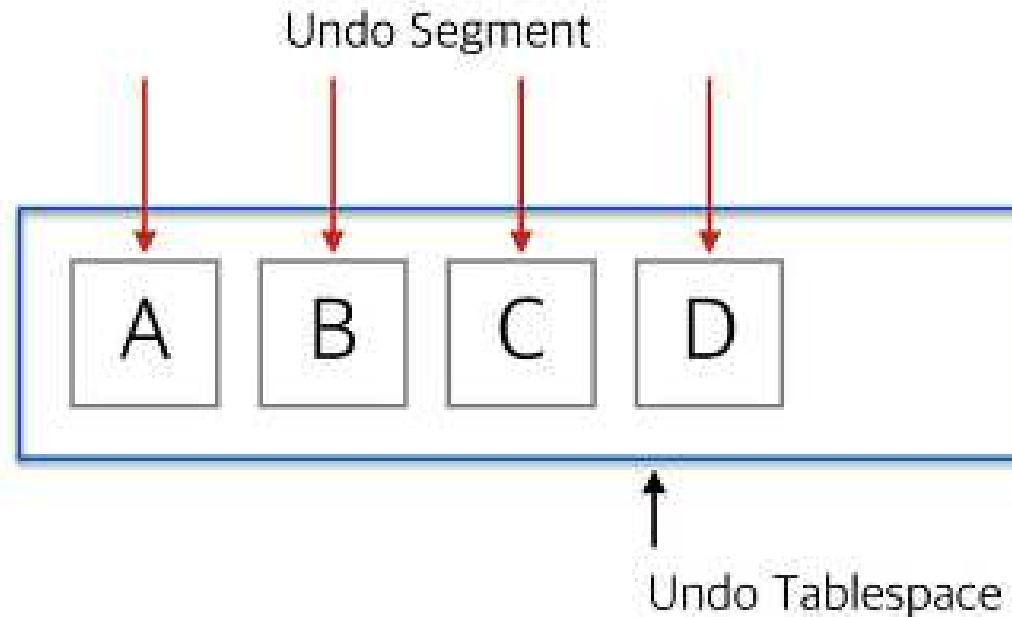
7. Tablespace & Datafiles

Read Consistency - 2



7. Tablespace & Datafiles

- Undo Segment 할당원리
 1. 현재 아래와 같이 4개의 Segment 할당되어 4개의 사용자가 사용 중

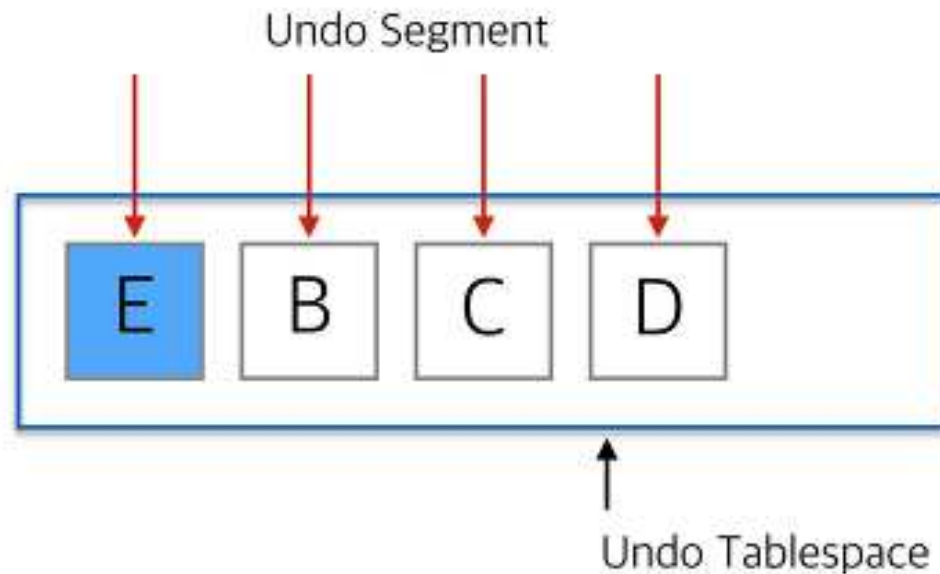


7. Tablespace & Datafiles

- Undo Segment 할당원리
 2. 새로운 사용자 E가 DML 수행
 3. E의 Server Process가 Undo Segment 할당을 위해 기존 Segment중에 재사용할 것이 있는지 찾음
 - 새로운 세션의 DML이 발생할 때마다 신규로 Undo Segment를 할당 받지 않음
 - 기존 Segment는 Commit이나 Rollback이 되면 재사용 가능한 상태가 되고 즉시 Segment의 내용이 flush 되는 것이 아니고 재사용되면서 flush

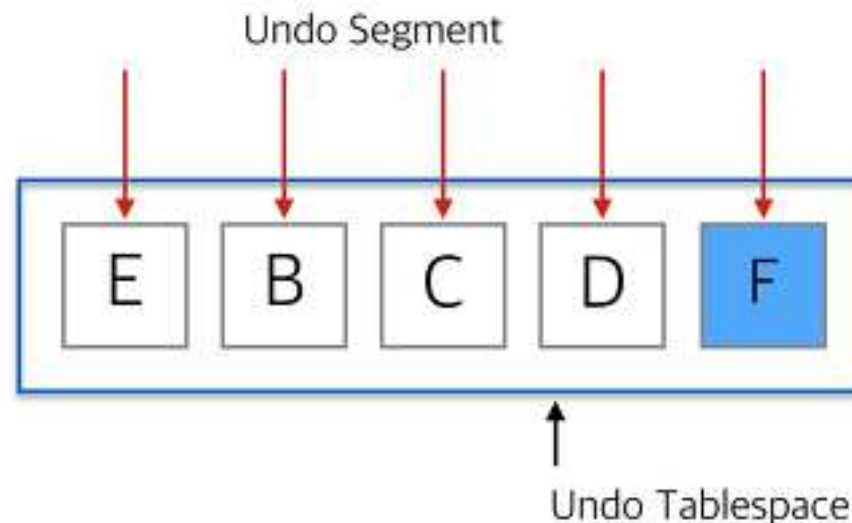
7. Tablespace & Datafiles

- Undo Segment 할당원리
 4. A가 그 시점에 Commit 수행
 5. E는 A가 쓰고 있던 Segment를 재사용



7. Tablespace & Datafiles

- Undo Segment 할당원리
 6. 새로운 사용자 F가 접속해 DML 수행
 7. 아무도 Commit이나 Rollback을 하지 않음
 8. F는 새로운 Segment를 할당 받음



7. Tablespace & Datafiles

- Undo Segment 할당원리
 9. 계속해서 새로운 Segment를 할당 받다가 Data file의 허용범위까지 늘어나면 하나의 Segment를 2개 세션 이상의 Undo Data를 함께 기록
 10. 그 이후에도 공간이 없으면 해당 트랜잭션은 에러 발생 -> Rollback

7. Tablespace & Datafiles

- Undo Segment 특징
 - Commit을 수행해도 Undo Segment안에 Undo Data는 지워지지 않고 남아있기 때문에 해당 Data file 사이즈를 줄일 수는 없음
 - Commit 수행 시 다른 서버 프로세스가 덮어 쓸 수 있게 해주는 것일 뿐 Undo Segment 안의 자료를 지우는 것은 아님
 - Undo Tablespace 크기가 비정상적으로 작을 경우에는 관리자가 다른 큰 Undo Tablespace를 신규로 만들고, Undo Tablespace를 신규 Undo Tablespace로 변경시킨 후 기존 Undo Tablespace를 삭제해 주어야 함

7. Tablespace & Datafiles

Undo 사용 현황 조회하기

1번 터미널

```
SQL> update emp set sal=5000 where empno=7902 ;
```

undo 사용량(record, block) 조회

```
SQL> select a.sid, a.serial#, a.username, b.used_urec, b.used_ublk  
       from v$session a, v$transaction b  
       where a.saddr = b.ses_addr ;
```

각 세션 별로 사용중인 Undo Segment 확인하기

```
SQL> select s.sid, s.serial#, s.username, r.name "ROLLBACK SEG"  
       from v$session s, v$transaction t, v$rollname r  
       where s.taddr=t.addr and t.xidusn=r.usn ;
```

7. Tablespace & Datafiles

Undo tablespace 조회 및 신규 생성

Undo 관련 조회

```
SQL> show parameter undo;
```

Undo tablespace 신규 생성

```
SQL> create undo tablespace undo01  
      datafile '/u01/app/oracle/oradata/PROD/undo01.dbf' size 10M  
      autoextend on ;
```

```
SQL> select * from dba_data_files ;
```


7. Tablespace & Datafiles

3.3 Undo tablespace 관리

Undo tablespace 변경

```
SQL> alter system set undo_tablespace=undo01 ;
```

기존 Undo tablespace 삭제

```
SQL> drop tablespace undotbs1 ;
```

undo retention 변경

```
SQL> alter system set undo_retention=600 ;
```

7. Tablespace & Datafiles

- Undo 관련 주요 파라미터
 1. UNDO_RETENTION(초 단위)
 - COMMIT 수행 후에도 해당 UNDO SEGMENT 내의 데이터를 다른 서버 프로세스가 덮어 쓰지 못하도록 막아주는 시간
 - 그러나, Undo segment 여분이 있을 경우에만 적용되며, 만약 모든 Undo Segment가 사용 중일 경우에는 적용되지 않고 Undo Segment가 재사용되어진다
 - 관련 에러 : Snap shot too old

7. Tablespace & Datafiles

- Ora-01555 : Snap shot too old
 - 사용자 A : 1시간 정도 수행되는 특정 데이터의 집계작업을 수행 중
 - 사용자 B : 사용자 A가 집계중인 데이터를 update 한 후 commit할 경우
 - 사용자 A: 사용자 B가 변경전의 데이터를 Undo Segment에서 찾아서 집계하게 됨
 - 만약 B가 Commit 한 후 사용자 C가 사용자 B의 Undo Segment를 덮어 썼다면 사용자 A는 집계쿼리를 완성할 수 없어 에러 발생
 - 이때 발생하는 에러가 Snap shot too old

7. Tablespace & Datafiles



- expired, unexpired undo block 조회

```
SQL> select tablespace_name , status , sum(bytes)
      from dba_undo_extents
      group by tablespace_name, status
      order by 1 ;
```

7. Tablespace & Datafiles

- Undo 관련 주요 파라미터

- 2. UNDO_RETENTION_GUARANTEE

- undo_retention으로 설정된 값 만큼 Segment가 재사용 되지 않도록 보장해주는 파라미터
 - 즉 Undo Segment가 부족하면 undo_retention가 설정되어 있더라도 무시하고 재사용하는데 이를 재사용되지 않도록 보장함
 - commit을 수행해도 특정 기간 동안 Undo Segment를 재사용하지 않기 때문에 Undo Tablespace가 커짐
 - default : NOGUARANTEE
 - alter tablespace undotbs1 retention guarantee ;

7. Tablespace & Datafiles

- TEMP Tablespace
 - Sort Operation 시에 임시로 사용됨
 - 영구 테이블스페이스로 변경 안됨
 - 성능을 위해 사용자 별로 1개씩 할당 권장
 - default temporary tablespace 는 삭제 안 됨

7. Tablespace & Datafiles

TEMP Tablespace 생성

```
SQL> create temporary tablespace temp2  
2 tempfile '/home/oracle/oradata/testdb/temp02.dbf' size 10M ;
```

Default TEMP Tablespace 변경

```
SQL> alter database default temporary tablespace temp2 ;  
SQL> drop tablespace temp ;
```

TEMP Tablespace 삭제

```
SQL> drop tablespace temp ;
```