

<DB 구현>

1. Oracle Database 구조	<p>[Oracle Database 전체구조]</p> <p><Instance></p> <ul style="list-style-type: none"> -SGA : Shared pool, DB buffer cache, Redo log buffer -B/G proc : PMON, SMON, DBWR, LGWR, CKPT, ARC(option) <p><Database></p> <ul style="list-style-type: none"> -Datafiles, Controlfile, Redologfiles, Parameterfile, Passwordfile, Archived log files(option)
2. Oracle 관련 Software 설치	<ol style="list-style-type: none"> 1. Oracle 사이트에서 Software 다운로드 2. 압축풀기 3. 실행 파일 실행 : \$] ./runInstaller 4. Software 설치
3. Oracle Database 생성	<ol style="list-style-type: none"> 1. 자동 생성 - dbca 사용 2. 수동 생성 - sqlplus에서 명령어(create database) 사용
4. Oracle Enterprise Manager 실행	<ol style="list-style-type: none"> 1. 웹브라우저 실행 2. https://localhost:1158/em <div style="text-align: center;"> ↑ ↑ DB가 존재하는 Port# Server의 Host명(IP주소) </div>
5. Parameter관리 및 DB 시작/종료	<p>[Parameter file]</p> <ul style="list-style-type: none"> -위치 : \$ORACLE_HOME/dbs -이름 : spfileSID.ora -특징 : Dynamic file, Binary file -조회 : SQL> show parameter db_block_size -변경 : SQL> alter system set db_cache_size = 512M scope = memory both spfile; <p>[DB 시작 단계]</p> <ul style="list-style-type: none"> -Shutdown : DB가 종료된 단계 -Nomount : Parameter file이 읽혀진 단계 Instance가 시작(구성)된 단계 작업 => DB 생성, Controlfile 재생성 -Mount : Controlfile이 읽혀진 단계 작업 => Datafile, Redologfile 위치/이름 변경 전체 DB 복구, Archive log mode 변경 -Open : Datafile, Redologfile이 읽혀진 단계 일반 DB user가 접속 가능한 단계 DB가 정상적으로 켜진 단계 <p>-명령어</p> <pre>SQL> startup SQL> startup mount SQL> startup nomount SQL> alter database mount; SQL> alter database open;</pre> <p>[DB 종료 모드]</p> <ul style="list-style-type: none"> -shut abort : 비정상 종료모드 -shut immediate : 체크포인트 발생 후 DB종료 -shut transactional : 진행 중인 트랜잭션이 종료될 때까지 기다려주는 종료 모드 -shut normal : default, DB에 접속된 세션이 나갈 때까지 기다려주는 종료 모드

6. Datafile 관리하기	<p>[저장공간 관련 용어] DB -> Tablespace -> Segment -> Extent -> Block</p> <p>[저장공간 추가]</p> <p>1. 새 T/S 추가</p> <pre>SQL> create tablespace users datafile '/u01/app/oracle/user01.dbf' size 100M autoextend on next 10M maxsize 500M;</pre> <p>2. 기존 T/S에 Datafile 추가</p> <pre>SQL> alter tablespace user add datafile '/u01/app/oracle/user02.dbf' size 50M;</pre> <p>3. 기존 Datafile의 사이즈 늘리기</p> <pre>SQL> alter database datafile '/u01/app/oracle/user02.dbf' resize 200M;</pre> <p>[저장공간 삭제]</p> <pre>SQL> drop tablespace users including contents and datafiles cascade constraints;</pre>
7. DB User 관리하기	<p>[User 생성]</p> <pre>SQL> create user arron identified by abc12345 default tablespace users temporary tablespace temp quota 30M on users profile appuser;</pre> <p>[권한 부여/회수]</p> <p>1. system 권한</p> <pre>SQL> grant create table to demo with admin option;</pre> <pre>SQL> revoke create table from demo;</pre> <p>2. object 권한</p> <pre>SQL> grant select on employees to demo with grant option;</pre> <pre>SQL> revoke select on employees from demo;</pre>

8. 데이터 충돌	<p>[Lock]</p> <ul style="list-style-type: none"> -서로 다른 세션이 동일한 데이터를 동일한 시간에 동시에 변경작업하는 것을 막기 위해서 오라클이 변경 작업 시 관련 행 단위로 Lock을 걸고 작업함. -트랜잭션이 종료될 때 Lock 해제됨. -Lock이 있는 곳에 작업이 요청되면 큐에서 대기함. <p>[Lock 충돌 발생]</p> <ul style="list-style-type: none"> -원인 : 마무리되지 않은 트랜잭션이 남아있는 경우 긴 트랜잭션이 진행되는 경우 -DBA가 모니터링(v\$sqlsession)하고 해결해야함. -DBA가 특정 세션을 kill 시킬 수 있음. <pre>SQL> alter system kill session 'SID, Serial#' immediate;</pre>
9. Controlfile, Redologfile 관리 및 다중화	<p>[Controlfile 다중화]</p> <ol style="list-style-type: none"> 1. DB 추가 <pre>SQL> alter system control_files= '/u01/app/oracle/control01.ctl', '/u02/app/oracle/control02.ctl', '/u03/app/oracle/control03.ctl' scope=spfile;</pre> <ol style="list-style-type: none"> 2. DB 종료 <pre>SQL> shut immediate</pre> <ol style="list-style-type: none"> 3. OS 추가 <pre>\$] cp /u01/app/oracle/control01.ctl /u03/app/oracle/control03.ctl</pre> <ol style="list-style-type: none"> 4. DB 시작 <pre>SQL> startup</pre> <ol style="list-style-type: none"> 5. 확인 <pre>SQL> select name from v\$controlfile;</pre> <p>[Redologfile 다중화]</p> <ol style="list-style-type: none"> 1. Redo log group 추가 <pre>SQL> alter database add logfile group 4 ('/u01/app/oracle/redo4a.log', '/u01/app/oracle/redo4b.log') size 50;</pre> <ol style="list-style-type: none"> 2. Redo log member 추가 <pre>SQL> alter database add logfile member '/u01/app/oracle/redo3c.log' to group 1;</pre> <p>[Archive log mode(권장사항) 변경]</p> <ol style="list-style-type: none"> 1. DB 종료 <pre>SQL> shut immediate</pre> <ol style="list-style-type: none"> 2. DB mount 시킴 <pre>SQL> startup mount</pre> <ol style="list-style-type: none"> 3. Archive log mode 변경 작업 <pre>SQL> alter database archivelog;</pre> <ol style="list-style-type: none"> 4. DB open 시킴 <pre>SQL> alter database open;</pre> <ol style="list-style-type: none"> 5. 확인 <pre>SQL> archive log list</pre>

10. Object 생성 계획 수립	<p>[개념모델링] - ERD가 만들어지는 단계</p> <p>[논리모델링] - 구체화된 ERD, 테이블 인스턴스 차트 등이 만들어지는 단계</p>
11. Object 구현 및 무결성 판단	<p>[물리모델링] - 실제 DB에 구현하는 단계(create table ----;)</p>
12. Backup	<p>[백업 용어] -백업 전략 : 전체 DB bk, 부분 DB bk -백업 모드 : Online bk(=hot bk), Offline bk(=cold bk) -백업 유형 : Full bk, Incremental bk -백업 파일 유형 : image copy, backupset -백업&복구 도구 : User관리 방식(cp), RMAN 방식</p>
13. Recovery	<p>[Controlfile 손실 시 복구] (상황) 다중화 되어 있는데 특정 멤버가 손실된 경우 1. DB 종료 SQL> shut abort 2. 남아있는 멤버를 사용해서 손실된 멤버를 복구함 \$] cp --- 3. DB 시작 SQL> startup</p> <p>[Redologfile 손실 시 복구] (상황) 다중화 되어 있는데 특정 그룹의 특정 멤버 손실 1. 손실된 멤버를 DB로부터 삭제 SQL> alter database drop logfile member '/u01/app/oracle/redo2b.log'; 2. Group에 멤버를 새롭게 추가 SQL> alter database add logfile member '/u01/app/oracle/redo2b_1.log'</p> <p>[Datafile 손실 시 복구] (case1) Non-critical datafile 손실 시 복구 1. DB mount SQL> startup mount 2. 손실된 Datafile만 Offline 시킴 SQL> alter database datafile 4 offline; 3. DB open SQL> alter database open; 4. restore \$] cp /backup/user01.dbf /u01/app/oracle/user01.dbf 5. recover SQL> recover datafile 4;</p> <p>(case2) Critical datafile 손실 시 복구 1. DB mount SQL> startup mount 2. restore \$] cp /backup/user01.dbf /u01/app/oracle/user01.dbf 3. recover SQL> recover database;</p>