

1. 오라클 데이터베이스 구조 탐색

오라클 DB 서버 구조

인스턴스 : physical database 내의 물리적인 data를 입/출력 할 수 있도록 해주는 하나의 소프트웨어로 메모리와 프로세스로 구성되어 있다.

- 메모리
 - System global area (SGA) : 데이터를 읽거나 변경하기 위해 사용하는 공용 메모리 영역
 - Shared pool : 오라클 DB를 운영하는데 필요한 정보를 저장하는 메모리 영역
 - 데이터베이스 버퍼 캐시 : 데이터 파일에서 읽은 데이터 블록 복사본을 보관
 - 리두 로그 버퍼 : DDL, DML 문장에 의해 DB에 변경되는 사항을 저장하는 메모리 영역
 - Large pool : SGA 내부의 대규모 메모리 할당을 제공하기 위한 선택적인 공간
 - Java pool : Java 프로그램 실행을 돕기 위한 메모리 공간
 - Streams pool : 데이터 공유, 복제 톨
 - Program global area (PGA) : DB에 접속하는 모든 유저에게 할당되어 각각의 서버 프로세스가 독자적으로 사용하는 메모리 영역
 - Stack space : 바인드 변수(Bind Variable) 사용 시 해당 바인드 변수를 저장
 - User global area (UGA) : 사용자 세션과 관련된 메모리 영역, DB 접속 방식에 따라 SGA또는 PGA 내에 존재
- 프로세스
 - 유저 프로세스 : 오라클 DB에 연결하는 응용 프로그램 또는 도구
 - 데이터베이스 프로세스
 - 서버 프로세스 : 유저 프로세스의 요청을 처리하는 프로세스
 - 백그라운드 프로세스
 - DBWn : 데이터베이스 버퍼 캐시의 수정된(더티) 버퍼를 디스크에 기록
 - LGWR : 리두 로그 버퍼를 디스크의 리두 로그 파일에 기록
 - CKPT : DBWn에 체크포인트 신호를 주고, control file과 datafile header에 해당 체크포인트 정보를 기록
 - SMON : 인스턴스 비정상 종료 시 Redo log file을 참조하여 recovery를 수행
 - PMON : 비정상 종료된 데이터베이스의 접속을 정리
 - RECO : 분산 데이터베이스에서 네트워크 또는 시스템 장애로 실패한 분산 트랜잭션 오류를 자동으로 처리
 - LREG : oracle net 리스너를 사용하여 데이터베이스 인스턴스 및 디스패처 프로세스에 대한 정보 등록
 - ARCn : 로그 스위치가 발생한 후에 리두 로그 파일을 지정된 장치로 복사
 - Daemon 과 응용프로그램 프로세스

데이터베이스

- Data file : 실제 data를 저장하는데 사용되는 파일
- Control file : DB를 유지, 관리 하기 위한 내부 정보를 저장하고 있는 파일
- Online redo log file : DB에서 발생한 모든 변경사항을 기록하는 파일

DB의 논리적/물리적 구조

논리적 구조

- Block : 오라클 DB에 data가 저장되는 최소 단위

- 오라클 DB의 I/O 단위
 - Block의 크기는 2, 4, 8, 16, 32 KB가 제공되며 기본 크기는 8 KB이다.
- Extent : 연속적인 Block들의 집합
 - Segment의 공간 할당 단위
- Segment : 하나 이상의 Extent로 구성
- Tablespace : 하나 이상의 Data File로 구성되며, 논리적으로는 여러개의 Segment가 모여서 구성
 - o SYSTEM Tablespace : DB 운영에 필요한 기본 정보를 담고 있는 Data Dictionary Table이 저장되는 공간
 - o SYSAUX Tablespace : SYSTEM Tablespace에 있는 다양한 유틸리티 및 기능들을 분리하여 저장하는 공간
- Database : 하나 이상의 Tablespace로 구성

물리적 구조

- Data file : 실제 data를 저장하는데 사용되는 파일
 - 논리적 저장구조인 Tablespace는 하나 이상의 Data file로 구성

오라클 DB Instance 구성

Single server 구성 : 하나의 DB에 하나의 instance가 할당되는 구성

- instance 장애 발생 시 storage 데이터에 접근할 수 없다.

High Availability (HA) 구성 : 같은 장비를 두개 구축하여 이중화 하는 방식

- Active-Standby로 구성하여 Active서버에 문제 발생 시 Standby서버가 Active로 전환
- 동일한 장비를 구축하는데 많은 비용이 발생
- Standby 서버의 자원 낭비
- Standby 서버가 동작할 때 데이터 유실이 발생할 수 있다.

Oracle Parallel Server (OPS) : 하나의 DB에 두개의 instance로 구성

- Storage를 공유하기 때문에 다른 instance에서 같은 데이터를 조회, 변경 할 수 있다.
- 특정 instance에서 변경된 데이터가 다른 instance에 적용될 때 Disk를 통해 데이터를 전송하므로 성능 저하 (RAC Ping)

Real Application Cluster (RAC) : 하나의 DB에 여러개의 instance로 구성

- OPS의 RAC Ping문제를 개선하여 성능이 크게 향상
- 특정 instance에서 변경된 데이터를 디스크 경유 없이 바로 다른 instance로 전송한다. (Cache Fusion)
- Public IP : RAC 유지/보수를 위해 관리자가 사용하는 망
- Private IP (Interconnet) : 각 instance를 연결하는 망
- Service IP (VIP) : 서비스 제공 망

DB Instance 연결

Connection : 유저 프로세스와 오라클 서버 프로세스간의 물리적 연결

Session : 사용자가 현재 데이터베이스 인스턴스에 아이디와 패스워드를 입력하고 로그인한 상태

2. 오라클 데이터베이스 관리 도구

오라클 12c 설치 (학습용)

- <https://edelivery.oracle.com> 접속
- Sign In
- Oracle Database 12c Enterprise Edition 검색 후 아래 이미지의 링크 클릭

DLP: Oracle Database 12c Enterprise Edition 12.1.0.2.0 (Oracle Database Enterprise Edition)

- 우측 상단의 View Items / Continue 클릭

<input type="checkbox"/> Download Queue	Terms and Restrictions	Platforms / Languages	Size	Published Date
<input type="checkbox"/> Oracle Database 12c Enterprise Edition 12.1.0.2.0	Oracle Standard Terms and Restrictions			Jul 03, 2018
<input checked="" type="checkbox"/> Oracle Database 12.1.0.2.0		Microsoft Windows x64 (64-bit)	2.6 GB	
<input type="checkbox"/> Oracle Database Client 12.1.0.2.0				
<input type="checkbox"/> Oracle Database Global Service Manager 12.1.0.2.0				
<input type="checkbox"/> Oracle Database Grid Infrastructure 12.1.0.2.0				
<input type="checkbox"/> Oracle Web Tier 11.1.1.7.0				

- Continue / Continue/ Download
- Oracle_SSN_DLM_07220855.exe 실행
- 파일 경로 설정 후 Next
- V47115-01_1of2.zip / V47115-01_2of2.zip 압축풀기
- databaseWsetup.exe 실행
- Oracle Database 12c 릴리스 1 설치 프로그램 - 단계 1/10
 - o My Oracle Support를 통해 보안 갱신 수신(W) 체크 해제
- Oracle Database 12c 릴리스 1 설치 프로그램 - 단계 2/10
 - o 데이터베이스 소프트웨어만 설치(D)
- Oracle Database 12c 릴리스 1 설치 프로그램 - 단계 3/10
 - o 단일 인스턴스 데이터베이스 설치(S)
- Oracle Database 12c 릴리스 1 설치 프로그램 - 단계 6/11
 - o Windows 내장 계정 사용(L)
- 설치 완료 후 Database Configuration Assistant 실행
- Database Configuration Assistant - 시작 - 단계 1/6
 - o 데이터베이스 생성(A)
- Database Configuration Assistant - 데이터베이스 생성 - 단계 1/14
 - o 고급 모드(R)
- Database Configuration Assistant - 데이터베이스 생성 - 단계 3/14

- 범용 또는 트랜잭션 처리
- Database Configuration Assistant - 데이터베이스 생성 - 단계 4/14
 - 전역 데이터베이스 이름(A) : orcl
 - SID(C) : orcl
- Database Configuration Assistant - 데이터베이스 생성 - 단계 6/14
 - 모든 계정에 동일한 관리 비밀번호 사용(C) : oracle_4U
- Database Configuration Assistant - 데이터베이스 생성 - 단계 7/15
 - 새 리스너 생성(C)
 - 리스너 이름(D) : LISTENER
 - 리스너 포트(E) : 1521
- Database Configuration Assistant - 데이터베이스 생성 - 단계 9/15
 - 샘플 스키마(E)
- Database Configuration Assistant - 데이터베이스 생성 - 단계 10/15
 - 메모리(C)
 - 메모리 크기(SGA 및 PGA)(J) : 1024 MB
 - 자동 메모리 관리 사용(K)

SQL Developer

- <https://otn.oracle.com> 접속



Downloads

Database Downloads

Java Downloads

SQL Developer

Oracle Instant Client

Oracle WebLogic Server

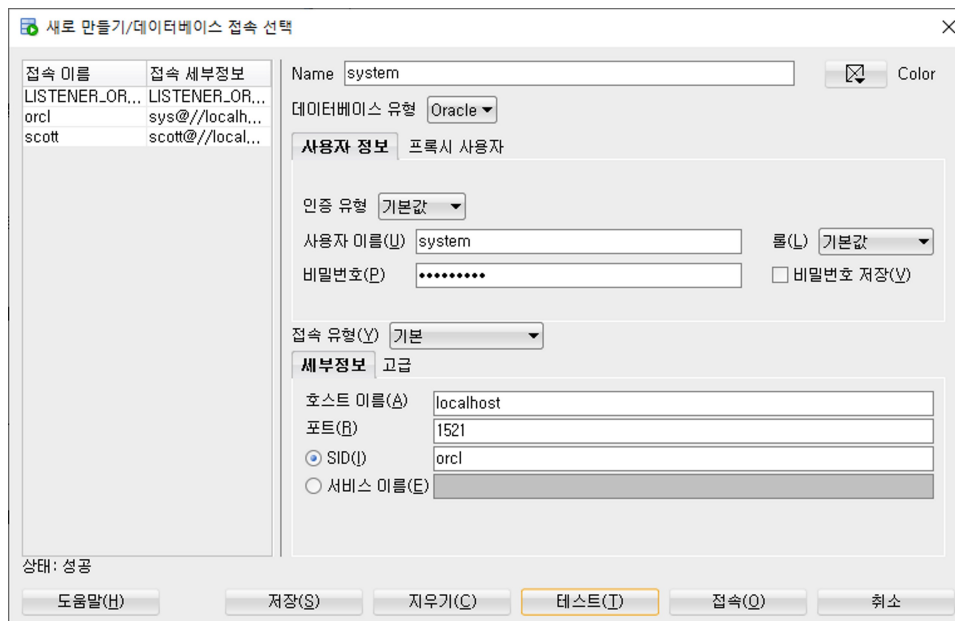
All Downloads

- Windows 64-bit with JDK 8 included Download

SQL Developer 환경설정

- 도구 - 환경설정 - 코드 편집기 - 글꼴
- 도구 - 환경설정 - 데이터베이스 - NLS 날짜 형식(F) : YYYY/MM/DD
- 도구 - 환경설정 - 데이터베이스 - 객체 뷰어 - 한 번 눌러 객체 열기(O) 체크 해제

DB 접속



```
create user oral  
identified by oracle_4U;
```

```
grant connect,  
resource,  
create view,  
create synonym,  
unlimited tablespace to oral;
```

새로 만들기/데이터베이스 접속 선택

접속 이름	접속 세부정보
LISTENER_OR...	LISTENER_OR...
orcl	sys@//localh...
scott	scott@//local...
system	system@//loc...

Name: ORA1 Color

데이터베이스 유형: Oracle

사용자 정보 프록시 사용자

인증 유형: 기본값

사용자 이름(U): ORA1 롤(L): 기본값

비밀번호(P): ☒ 비밀번호 저장(V)

접속 유형(Y): 기본

세부정보 고급

호스트 이름(A): localhost

포트(P): 1521

☒ SID(I): orcl

☐ 서비스 이름(E):

상태:

도움말(H)
저장(S)
지우기(C)
테스트(T)
접속(O)
취소

setup.sql 실행

3. 데이터베이스 Instance 관리

초기화 파라미터 파일

초기화 파라미터 파일 : 인스턴스가 시작될 때 필요한 설정들에 대한 파라미터 값을 가지고 있는 파일

- SPFILE (Sever Parameter File)
 - 2진 Binary 파일로 구성되어 있으며, OS에서 수정이 불가능
 - ALTER 문을 통해 DB운영 중에도 파라미터를 수정 할 수 있고 서버를 재시작 하지 않아도 적용
 - 기본 파일명 : spfile<SID>.ora
- PFILE (Parameter File)
 - Text 형태로 구성되어 있으며, OS에서 수정이 가능 (수정 후 DB를 재시작해야 적용)
 - 기본 파일명 : init<SID>.ora

초기화 파라미터 값 변경

파라미터

- Static parameter
 - 파라미터 파일에서만 수정 가능
 - 인스턴스를 재시작해야 적용
- Dynamic parameter
 - DB운영 중에도 파라미터 수정 가능
- 파라미터 변경시 SCOPE 옵션
 - SCOPE 옵션
 - o Memory : 변경 사항이 메모리에만 적용, DB가 재시작 되면 변경 이전 값으로 돌아감
 - o Spfile : 변경 사항이 spfile에만 저장되고 현재 DB에는 영향을 미치지 않음
 - Static parameter은 spfile 옵션만 지정 가능 (DB를 재시작해야 적용 가능)
 - o Both : 변경 사항이 메모리와 spfile에 모두 저장, 현재와 DB 재시작 후 모두 영향을 미침

오라클 시작 및 종료

오라클 시작

- SHUTDOWN : 오라클이 완전 정지 된 상태
- NOMOUNT : 파라미터 파일에 접근하여 Instance를 실행
- MOUNT : NOMOUNT 상태의 파라미터 파일에 지정된 Control file을 실행
- OPEN : Control file을 읽어 Data file과 Redo log file의 정보를 확인하고 모든 파일 open

오라클 종료

- ABORT : 강제종료, 다음 오라클 시작시 인스턴스 복구가 필요
- IMMEDIATE : commit된 데이터를 Data file에 저장하고 commit되지 않은 데이터를 rollback 시킨 후 인스턴스 종료
- TRANSCITIONAL : 새로운 트랜잭션을 할 수 없으며, 모든 트랜잭션이 종료되면 인스턴스 종료
- NOMAL : 새로운 사용자의 접속을 허용하지 않으며, 모든 사용자가 세션을 종료하면 인스턴스 종료

종료모드	ABORT	IMMEDIATE	TRANSCITIONAL	NOMAL
새로운 연결 허용	X	X	X	X
현재 세션 종료 시까지 대기	X	X	X	O
현재 트랜잭션 종료 시까지 대기	X	X	O	O
체크포인트 적용 및 파일 닫기	X	O	O	O

4. 오라클 네트워크 환경 구성

Oracle net 리스너

리스너 : 네트워크를 이용하여 클라이언트에서 오라클 서버로 연결하기 위한 오라클 네트워크 관리자

- 파일 형식 : listener.ora

이름 지정 방식

이름 지정 방식을 이용하여 원하는 데이터베이스로 접속할 수 있는 정보를 획득

Easy connect : 대상 데이터베이스의 TCP/IP를 이용하여 대상 데이터베이스에 접속하는 방식

- 기본적으로 활성화되며 클라이언트의 구성이 필요 하지 않음
- TCP/IP만 지원
- 고급 연결 옵션 지원하지 않음

로컬 이름 지정 : 클라이언트에 존재하는 tnsnames.ora 파일에 등록된 대상 데이터베이스의 정보를 이용하여 접속하는 방식

- 클라이언트에 tnsnames.ora 파일이 필요
- 모든 oracle net 프로토콜 지원
- 고급 연결 옵션 지원

디렉토리 이름 지정 : 별도로 존재하는 디렉토리 서버에서 대상 데이터베이스의 정보를 모두 관리하는 방식

- tnsnames.ora 파일과 함께 LDAP 필요
- 고급 연결 옵션 지원

외부 이름 지정 : 비오라클 방식으로 오라클과 호환되는 서드파티 도구나 응용프로그램에서 사용

Dedicated server 와 Shared server 구성 비교

Dedicated server : 유저프로세스와 서버 프로세스가 1:1로 연결되는 환경

- 서버프로세스 각각의 PGA가 할당된다.
- 유저프로세스와 서버프로세스가 1:1로 연결되기 때문에 좋은 성능을 가진다.
- OLTP환경에서 자원이 낭비된다.

Shared server : 다수의 유저프로세스가 소수의 서버프로세스를 공유해서 사용하는 환경

- Dispatcher를 사용하여 자원낭비를 줄일 수 있다.

- Dispatcher : 클라이언트에서 보낸 요청을 받아 request queue에 전달하는 역할
- Shared server환경에서의 UGA(User Global Area)는 SGA 내부에 생성 된다.

- Shared server 구성 시 고려 사항

Shared server를 사용하여 수행하지 않아야 하는 특정 데이터베이스 작업유형

- 데이터베이스 관리
- 백업 및 복구 작업
- 일괄 처리 및 대량 로드 작업
- 데이터 웨어하우스 작업

5. 유저 보안 관리

데이터베이스 계정

유저 계정

각 데이터베이스 유저 계정에는 아래의 내용이 포함되어 있다.

- 고유 username
- 인증 방식
- 기본 테이블스페이스
- 임시 테이블스페이스
- 유저 프로필
 - 프로필 : 리소스의 소비를 제어하고 계정 상태 및 암호 만료를 관리하기 위해 생성하는 하나의 오브젝트
- 초기 Consumer Group
- 계정 상태

스키마 : 데이터베이스 유저가 소유하는 데이터베이스 객체의 모음

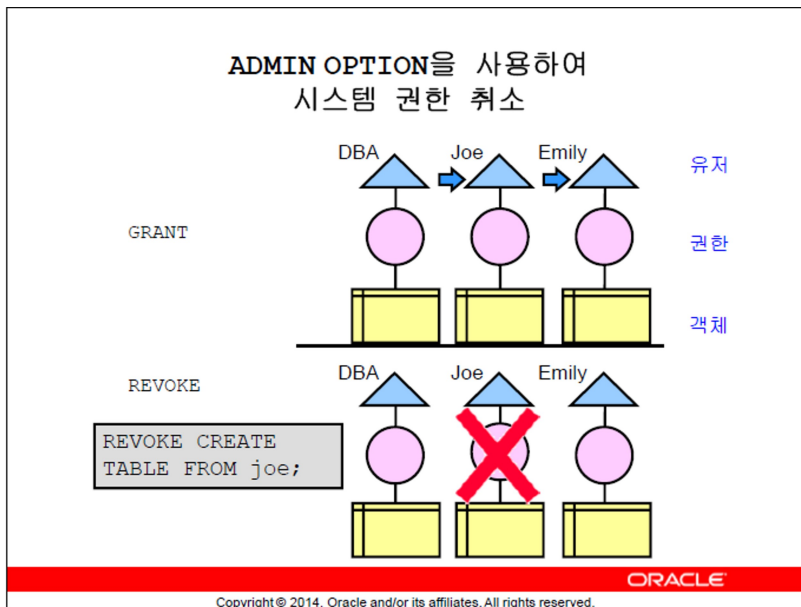
기본적으로 생성되는 관리 계정

- SYS : Oracle DB 관리자로 Super user이다. Oracle 시스템의 기반이 되는 Data dictionary 소유자이며 DB 생성과 삭제 가능
 - SYSDBA의 권한을 가짐
- SYSTEM : SYS와 유사한 권한을 가지고 있지만 DB 생성과 삭제가 불가능, 생성된 DB를 운영 하기 위한 권한을 갖는 계정
 - SYSOPER의 권한을 가짐
- SYSBACKUP : Oracle RMAN 백업 및 Recovery 작업을 지원
- SYSDG : Oracle Data Guard 작업을 지원
- SYSKM : Transparent Data Encryption 전자 지갑(wallet) 작업을 지원

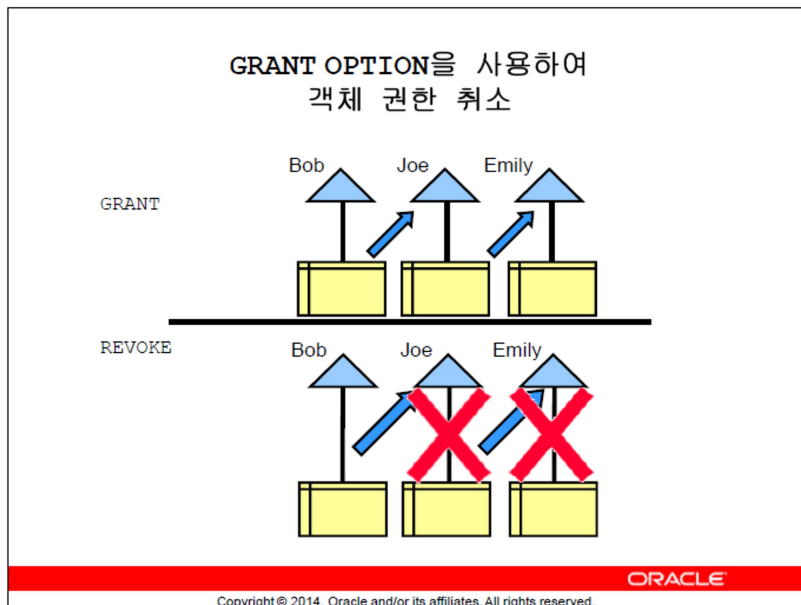
권한

유저 권한

- 시스템 권한 : 유저가 DB에서 특정 작업을 수행할 수 있도록 한다.
 - WITH ADMIN OPTION을 사용하여 시스템 권한 취소



- 객체 권한 : 사용자가 특정 객체를 access 및 조작할 수 있도록 한다.
- WITH GRANT OPTION을 사용하여 객체 권한 취소



관리 권한

- SYSDBA : Oracle 시스템에서 모든 권한을 가지며, DB 생성과 삭제가 가능하다.
- SYSOPER : DB를 운영 관리하기 위한 권한으로 DB 생성과 삭제가 불가능하며, 다른 유저 소유의 DB에는 접근할 수 없다.
- SYSBACKUP : Oracle RMAN 또는 SQL Plus를 사용하는 RMAN 백업 및 Recovery 작업
- SYSDG : Data Guard Broker 또는 DGMGRL 명령행 인터페이스를 사용하는 Data Guard 작업
- SYSKM : Transparent Data Encryption 전자 지갑(wallet) 작업 관리

인증

유저 인증

- Password 인증 : 오라클 데이터베이스에 의한 인증
 - Password File을 사용하여 DB 계정 및 패스워드를 확인

- External 인증 : 데이터베이스 외부의 방식으로 인증
 - o 운영체제 : OS에 의한 인증을 의미하며 DB 계정 및 패스워드를 확인하지 않고 쉽게 DB에 접근 가능
 - o kerberos
 - o radius
- Global 인증 : LDAP 기반 디렉토리 서비스를 사용하여 인증

관리자 인증

- 운영체제 보안
 - DBA는 파일을 생성하고 삭제하는 OS권한을 가져야 한다.
 - 일반 유저는 파일을 생성하고 삭제하는 OS권한을 가질 수 없다.
- 관리자 보안
 - SYSDBA 및 SYSOPER 연결
 - Password File 인증 방식에서는 DBA 유저 이름에 대한 감사 수행
 - OS인증에서는 OS계정 이름에 대한 감사 수행
 - OS인증이 Password File인증보다 우선순위가 높음
 - Password File은 대소문자를 구분하는 암호를 사용

롤(Role)을 사용하여 권한 관리

롤(Role) : 하나 이상의 권한으로 이루어진 집합체

- 롤에 권한 부여/취소 가능
- 롤을 유저나 다른 롤에 부여/취소 가능
- 시스템 및 객체 권한으로 구성
- 롤을 활성화 하려면 암호가 필요할 수 있음
- 롤은 유저가 소유하지 않으며 스키마에 존재하지 않음

롤 사용 시의 이점

- 권한 관리 용이 : 유저에게 롤을 부여한 후 롤에 권한을 부여
- 동적 권한 관리 : 롤에 관한 권한을 수정하면 그 롤을 부여받은 유저는 즉시 수정된 권한을 얻는다.
- 권한의 선택적 가용성 : 롤을 활성화/비활성화 하여 권한을 일시적으로 설정/해제 가능

기본적으로 생성되는 롤

- CONNECT ROLE : 사용자가 DB에 접속 가능하도록 하기 위해서 가장 기본적인 시스템 권한들이 부여된 롤
- RESOURCE ROLE : 사용자가 객체(테이블, 뷰, 인덱스)를 생성할 수 있도록 하기 위한 시스템 권한들이 부여된 롤
- DBA ROLE : 모든 시스템 권한이 부여된 롤

6. 데이터베이스 저장 영역 구조 관리

데이터베이스 블록

블록 : 오라클 DB에 data가 저장되는 최소 단위 (논리적)

- 블록 헤더 : 블록의 내부 정보를 저장하며, 위에서 아래로 확장
 - o 데이터 블록 주소
 - o 세그먼트 유형
 - o 테이블 디렉토리
 - o 행 디렉토리
 - o 트랜잭션 슬롯
- 행 데이터 : 블록에 있는 행의 실제 데이터
 - 블록은 처음 구성될 때 블록 헤더와 사용 가능 영역으로만 구성되어 있으며, 데이터가 Insert될 때마다 아래에서 위로 확장
- 사용 가능 영역 : 블록의 가운데 위치하며, 필요에 따라 헤더와 행 데이터의 확장 공간이 될 수 있다.

테이블스페이스 설정

TABLESPACE TYPE

- Permanent : 영구히 데이터를 저장하는 일반적인 테이블스페이스
 - o SYSTEM : DB운영에 필요한 기본 정보를 담고 있는 Data dictionary table 이 저장되는 공간
 - o SYSAUX : SYSTEM 테이블스페이스에 있는 다양한 유틸리티 및 기능을 분리하여 저장한 공간
- Temporary : 임시로 데이터를 저장하는 임시 테이블스페이스 (주로 sort 작업할 때 사용)
- Undo : rollback을 위한 undo 데이터를 저장하는 테이블스페이스

BIGFILE

- Bigfile : 하나의 테이블스페이스에 하나의 데이터 파일만 저장 가능
- Smallfile : 하나의 테이블스페이스에 여러개의 데이터 파일 저장 가능

EXTENT ALLOCATION

- Automatic : 테이블스페이스 내의 extent의 크기가 시스템에 의해 관리
 - 임시 테이블스페이스에 대해서는 Automatic 지정 불가
- Uniform : 사용자가 지정한 동일한 extent의 크기로 테이블스페이스 관리
 - Undo 테이블스페이스에 대해서는 Uniform 지정 불가

LOGGING

- Logging : 테이블스페이스의 객체에 대한 변경 내용을 리두 로그에 기록
- No logging : 리두 로그 생성이 억제되어 logging 테이블 보다 적은 내용만 리두 로그에 기록
 - 리두 로그가 전혀 생성 되지 않는 것은 아니다.

SEGMENT SPACE MANAGEMENT

- Automatic : 비트맵을 이용하여 세그먼트 내의 영역 관리
 - ASSM(Automatic Segment Space Management)을 사용하여 세그먼트 영역을 자동으로 관리
- Manual : Freelist를 이용하여 사용 가능 영역을 수동으로 관리
 - Freelist : Insert할 공간이 있는 데이터 블록 리스트

COMPRESSION

- Basic : 데이터베이스 블록내의 데이터들을 압축
 - Compress 장/단점
 - 장점
 - 저장 공간 절약
 - I/O 감소로 성능 향상 및 버퍼캐시 사용 효율 증대
 - SELECT문의 성능 향상
 - 단점
 - CPU 사용률 증가
 - UPDATE문의 성능 저하
- None
- OLTP

OMF (Oracle Managed Files)

OMF (Oracle Managed Files) : 오라클 DB를 구성하는 운영체제 파일을 오라클 서버에 의해 자동으로 관리

파라미터	설명
DB_CREATE_FILE_DEST	데이터 파일 및 임시 파일에 대한 디렉토리의 위치를 정의
DB_CREATE_ONLINE_LOG_DEST_n	리두 로그 및 Control File 생성을 위한 위치를 정의
DB_RECOVERY_FILE_DEST	Fast Recovery Area의 기본 위치 정의

데이터베이스 확장

데이터베이스 확장 방법

- 새 테이블스페이스 생성
- 기존 Small File 테이블스페이스에 데이터 파일 추가
- 데이터 파일 크기 증가
- 데이터 파일의 동적 증가 제공

7. 공간 관리

블록 공간 관리

Row Migration : update문 수행 등으로 인해 로우 데이터의 크기가 커져 데이터 블록의 빈 공간에 들어갈 수 없게되었을 때 다른 데이터 블록에 로우 데이터가 저장 되는 현상

- Row Migration이 발생하면 기존 데이터 블록에는 새로운 데이터 블록에 대한 포인터 값이 남게 되고 원하는 데이터를 얻어 오려면 여러개의 블록을 스캔해야 하기 때문에 성능이 저하 된다.

Row Chaining : 하나의 로우 데이터의 크기가 블록의 데이터 저장 공간보다 클 때 발생하는 현상으로 하나의 로우 데이터를 분리 하여 서로 다른 블록에 저장한다.

- Row Migration과 마찬가지로 여러개의 블록을 스캔해야 원하는 로우 데이터를 얻을 수 있기 때문에 성능이 저하 된다.

PCTFREE : 데이터 블록에 실제 데이터가 Insert될 때 설정해준 값만큼 여유 공간을 만들며, Row Migration현상을 방지하기 위해 사용한다.

- PCTFREE의 기본값은 10(%) 이다.

익스텐트 관리

Extent management 방식

- Dictionary 방식 : extent정보를 시스템 테이블스페이스에 저장된 데이터 디렉터리에서 관리
 - 테이블스페이스에 존재하는 extent의 정보가 변경될 때마다 데이터 디렉터리의 해당 extent 정보를 갱신해야 한다.
 - Segment마다 각각 다른 extent size 설정이 가능
 - 데이터 디렉터리 테이블에서 undo가 발생
 - 모든 segment가 소수의 디렉터리 테이블을 조회하고 갱신해야 하므로 데이터 디렉터리 테이블에 대한 경합이 발생할 가능 성이 높다.
 - 주기적으로 extent를 병합하거나 처음부터 extent크기를 동일하게 구성해야한다.
- Local 방식 : extent정보를 bitmap을 이용하여 테이블스페이스의 header에서 관리
 - 데이터 디렉터리 테이블을 갱신하거나 참조하지 않는다. 따라서 데이터 디렉터리 테이블에 대한 경합이 감소한다.

Extent 크기 조정

- Autoallocate : 테이블스페이스 내의 extent의 크기가 시스템에 의해 관리
 - 임시 테이블스페이스에 대해서는 Automatic 지정 불가
- Uniform : 사용자가 지정한 동일한 extent의 크기로 테이블스페이스 관리

- Undo 테이블스페이스에 대해서는 Uniform 지정 불가

세그먼트 유형

Segment : 하나 이상의 extent로 구성

- Object 중에서 저장 공간을 가지고 있는것

Segment의 종류

- Data Segments : table, table partition, cluster
- Index Segments : index-organised table, index partition
- Undo Segments : rollback segment
- Temporary Segments

세그먼트 공간 관리

FLM (Free List Management) : Freelist를 이용하여 사용 가능 영역을 수동으로 관리

- 프리 블록의 여부는 PCTFREE의 속성에 의해 결정
- 세그먼트 헤더 블록의 Freelist에서 프리 블록의 시작과 끝을 관리
- 개별 데이터 블록에는 블록이 Freelist에 존재하는지의 여부와 다음 프리 블록의 위치 정보를 저장
- 블록의 지속적인 I/O 작업을 반복하기 때문에 비효율적

ASSM (Automatic Segment Space Management) : 비트맵을 이용하여 세그먼트 내의 영역을 자동으로 관리

Deferred_segment_creation

Deferred_segment_creation : 테이블을 생성시 데이터가 들어오기 전까지 세그먼트를 만들지 않게하는 파라미터

- Default로 설정되어 있다.

세그먼트 축소

SHRINK : Segment에 할당된 Extent를 Tablespace로 반환하는 작업

- Shrink 작업으로 Segment를 축소 (Tablespace의 용량은 변하지 않음)
- Insert, Delete를 통해 데이터를 이동하기 때문에 추가적인 공간이 필요하지 않음
- 일부 데이터만 이동하기 때문에 Row Migration이나 Chaining이 전체적으로 해결되지 않음
- Shrink 명령어
 - ALTER TABLE <테이블명> SHRINK SPACE [COMPACT] [CASCADE];
 - COMPACT
 - Shrink 명령어 뒤에 COMPACT를 명시하면 shrink작업은 진행하지만 HWM는 재조정하지 않고, 공간 또한

즉시 반환하지 않는다.

- ALTER TABLE ~ SHRINK SPACE 명령을 사용하여 HWM을 재조정하고 단편화된 공간을 회수해야 한다.
- HWM에 대한 정보를 바꾸는 것은 온라인성 작업이 아니기 때문에 두 단계로 나뉨
 - HWM(High Water Mark) : 마지막까지 등록된 블록 위치
- CASCADE
 - Shrink 작업의 대상인 테이블과 관련된 모든 오브젝트들도 Shrink작업을 수행

테이블 압축

Basic : Direct path I/O 작업의 경우에만 압축 가능

- Bulk loading 작업이 빈번하게 일어날 때 유리
- 압축 과정
 - o 압축이 되지 않은 상태로 데이터 삽입
 - o PCTFREE에 도달하면 압축 진행
 - o 다시 압축이 되지 않은 상태로 데이터 삽입
 - o PCTFREE에 도달하면 압축 진행
 - o 위의 과정 반복

Advanced : 모든 DML작업에 대해 압축 가능

- OLTP환경에 적합
- ACO (Advanced Compression Option) 옵션을 구매해야 정식으로 사용 가능
- 반복되는 데이터를 블록 헤더에 한번만 저장하는 방식

8. 언두 데이터 관리

언두 데이터

언두 데이터 : 사용자가 DML을 수행할 경우 발생하는 원본 데이터

- 데이터 조작어를 통해 수정되기 전 원래 데이터의 복사본
- 데이터를 변경하는 모든 트랜잭션에 대해 캡처
- 트랜잭션이 종료될 때까지 보존
- 지원하는 작업
 - o 롤백
 - o 읽기 일관성 및 Flashback Query
 - o 실패한 트랜잭션 Recovery

언두 세그먼트 : 언두 데이터를 저장하는 세그먼트

- 각 트랜잭션은 하나의 언두 세그먼트에만 할당
- 경우에 따라 하나의 언두 세그먼트로 여러개의 트랜잭션을 처리 가능

언두 테이블스페이스 : 언두 세그먼트를 저장하는 테이블스페이스

- 서버 프로세스가 직접 관리
- 단일 인스턴스와만 연관

언두 데이터 및 리두 데이터 비교

	언두	리두
기록 내용	변경 사항을 되돌리는 방법	변경 사항을 재생성하는 방법
사용 목적	롤백, 읽기 일관성, Flashback	DB 변경 사항 롤포워드
저장 위치	언두 세그먼트	리두 로그 파일

언두 관리

언두 관리를 통해 아래의 오류를 방지

- 언두 테이블스페이스의 공간 오류 (ORA-30036)
 - 언두 테이블스페이스 확장
 - 새로운 언두 테이블스페이스 생성

- 쿼리문의 로직을 작은 단위로 잘라서 실행
- "Snapshot too old" 오류 (ORA-01555)
 - 적절한 언두 retention 간격 구성
 - 언두 테이블스페이스의 크기를 적절히 조정
 - 언두 retention 보장을 고려

자동 언두 관리

- 언두 테이블스페이스에서 언두 데이터 및 언두 공간 관리 완전 자동화
 - 장기 실행 query를 충족하도록 Autoextend
 - Retention을 최적화하도록 고정 크기 테이블스페이스에서 자체 튜닝

UNDO_RETENTION : 이미 커밋된 언두 정보를 보존해야 하는 기간을 초 단위로 지정하는 파라미터

- UNDO_RETENTION 파라미터를 설정하는 경우
 - o Autoextend 옵션이 활성화된 경우
 - o LOB에 대해 언두 Retention을 설정할 경우
 - o Retention을 보장하려는 경우
- Status
 - o ACTIVE : 커밋되지 않은 언두 정보
 - o UNEXPIRED : 커밋된 언두 정보
 - o EXPIRED : UNDO_RETENTION이 만료 된 언두 정보

GUARANTEE 옵션

- GUARANTEE : 언두 테이블스페이스의 저장 공간이 부족해도 Retention 보장
 - ORA-01555 에러가 발생하지 않지만, ORA-30036이 발생할 가능성이 높아지고 트랜잭션 실패 가능성이 높다.
- NOGUARANTEE : 언두 테이블스페이스의 저장 공간이 부족할 때 UNEXPIRED를 덮어써 Retention을 보장하지 않음
 - ORA-01555 에러 발생률은 높아지고, ORA-30036 에러는 발생하지 않는다.

임시 언두

임시 언두 이점

- 언두 테이블스페이스에 저장되는 언두의 양을 감소
- 리두 로그의 크기를 감소
- Oracle Active Data Guard 옵션이 있는 물리적 Standby Database의 임시 테이블에서 DML작업을 수행 가능

9. 데이터 동시성 관리

LOCK

다중 세션에서 동일한 데이터를 동시에 변경하는 것을 방지
주어진 명령문에 대해 가능한 가장 낮은 레벨에서 자동으로 획득

LOCK 매커니즘

높은 레벨의 데이터 동시성을 제공

- 행 레벨의 Lock 획득
- 일반적인 Select문 쿼리에서는 Lock을 걸지 않음

자동 큐 관리

트랜잭션이 종료(Commit, Rollback)될 때까지 Lock 보유

DML LOCK

각 DML 트랜잭션은 다음의 Lock 획득

- 갱신 중인 행에 대한 EXCLUSIVE 행 잠금
- ROW EXCLUSIVE(RX) 모드에서 DDL문을 실행하지 못하도록 테이블에 대한 테이블 Lock(TM) 획득

ENQUEUE 매커니즘

Lock은 자동으로 큐에 저장되며, Lock을 보유하는 트랜잭션이 완료되면 다음 트랜잭션이 Lock을 보유

Lock을 이미 보유한 세션은 해당 Lock의 변환 요청 가능

LOCK 충돌

Lock 충돌 원인

- Commit 되지 않은 변경 사항
- 장기 실행 트랜잭션
- 필요 이상으로 높은 Lock 레벨

Lock 충돌 해결

- Lock을 보유하는 세션을 커밋 또는 롤백
- Lock을 보유하는 세션을 종료 (Kill Session)

Deadlock

둘 이상의 세션이 각각 Lock 된 데이터를 서로 대기하고 있는 경우

- Oracle DB는 Deadlock을 자동으로 감지
- 트랜잭션 종료(Commit, Rollback)를 통해 해결

10. 오라클 데이터베이스 감사(Audit) 구현

데이터베이스 보안

보안 시스템은 내부 데이터의 기밀성을 보장

- 데이터 및 서비스에 대한 액세스 제한
- 유저 인증
- 의심스러운 작업 모니터

의심스러운 작업 모니터

감사 (Audit)

- 사용자의 행동을 감시하거나 DB에 관한 통계자료를 얻는 목적으로 사용

감사 형태

- 필수 감사 : 모든 오라클 데이터베이스는 다른 감사 옵션이나 파라미터에 관계없이 특정 작업을 감사한다.
- 표준 데이터베이스 감사 : 데이터베이스에 영향을 끼치는 작업을 감사하거나 특정 데이터베이스 작업에 대한 데이터를 모니터링하고 수집한다.
- 값 기준 감사 : 데이터베이스 감사 뿐만 아니라 DML 수행에 의해서 변경된 데이터까지 감사한다.
 - 데이터베이스 트리거를 통해 구현
- FGA (Fine-Grained Auditing) : 특정 사용자의 조회 및 DML 작업을 감사한다.

11. 백업 및 Recovery 개념

Database Fail 유형

Statement Failure : 데이터베이스 작업이 정상적으로 실행되지 않고 오류가 발생하는 경우

- 주 원인
 - 오타, 문법 오류
 - 권한이 부족한 상태에서 작업 수행
 - 저장 공간 부족
- 해결 방법
 - 오타 수정, 문법 수정
 - 적절한 객체 권한, 시스템 권한 부여
 - 저장 공간 확장

User Process Failure : 데이터베이스에 접속된 유저의 세션이 끊기는 경우

- 주 원인
 - 유저가 비정상적으로 연결은 끊은 경우
 - 유저 세션이 비정상적으로 종료된 경우
 - 세션을 종료시키는 프로그램의 오류
- 해결 방법
 - PMON이 커밋되지 않은 변경사항을 자동으로 롤백하고 Lock을 해제

Network Failure : 데이터베이스 접속이 되지 않는 경우 (네트워크 장애는 주로 네트워크 관리자나 업체에 문의)

- 주 원인
 - 리스너가 꺼져 있는 경우
 - 리스너가 비정상 종료된 경우
 - NIC에 문제가 발생한 경우
 - 네트워크에 문제가 있는 경우
- 해결 방법
 - 리스너를 키고 데이터베이스 재시작
 - 백업 리스너 구성
 - 다중 네트워크 카드 구성
 - 백업 네트워크망 구성

User error : 정상적으로 작업이 실행되지만 잘못된 작업을 수행한 경우

- 주 원인
 - 유저의 실수로 데이터를 삭제하거나 수정
 - 유저가 테이블을 삭제한 경우

- 해결 방법
 - 트랜잭션 롤백
 - Recycle bin에서 테이블을 복구 (Flashback 작업)
 - Flashback : 특정 시간 또는 특정 시점으로 되돌려 데이터를 복구하는 기술
 - Recycle bin이 이미 지워졌거나 PURGE 옵션으로 테이블을 삭제한 경우 PITR을 사용하여 복구

Instance Failure : 데이터베이스 Instance가 비정상 종료되는 경우

- 주 원인
 - Server PC 다운, 정전
 - 비상 종료 (SHUTDOWN ABORT, STARTUP FORCE)
 - 주요 백그라운드 프로세스 중 하나의 문제가 있는 경우
- 해결 방법
 - STARTUP 명령을 사용하여 Instance 재시작 (SMON이 자동으로 데이터베이스를 동기화 시킨다.)

Media Failure : 데이터베이스를 구성하는 필수 물리적인 파일 중 하나가 손상/손실 된 경우

- 주 원인
 - DB를 구성하는 필수 파일 삭제 또는 손상
- 해결 방법
 - 백업 파일을 사용하여 손상된 파일을 복구

Instance Recovery

CKPT(체크포인트) 프로세스

- DBWn이 SGA의 수정된 데이터 블록을 디스크에 기록한 내용을 Control File에 저장
- 로그 스위치가 발생할 경우 CKPT 프로세스는 체크포인트 정보를 데이터 파일의 헤더에 기록
 - 체크포인트 정보
 - 체크포인트 위치 : 리두 로그 파일에서 Instance Recovery가 시작될 위치
 - SCN (System Change Number) : 유저가 commit 할때 해당 트랜잭션에 부여되는 고유번호

리두 로그 파일

- 데이터베이스 변경 내용 기록
- 데이터 손실로부터 보호하려면 다중화 되어야 한다.

LGWR 프로세스 : 리두 로그 버퍼를 디스크의 리두 로그 파일에 기록

- LGWR 활동 주기
 - 커밋을 수행할 때
 - 리두 로그 버퍼의 1/3 이상 사용되었을 때
 - 1MB 이상의 리두 로그가 생성 되었을 때
 - DBWn에 의해 변경된 데이터 블록을 저장하기 전
 - 리두 로그 스위치가 발생 하였을 때
 - 3초 마다

Instance Recovery 특징

- 종료 시 동기화되지 않은 파일이 있는 데이터베이스를 열려고 할 때 발생
- 리두 로그 그룹에 저장된 정보를 사용하여 파일 동기화

Instance Recovery 단계

- Instance 시작 (데이터 파일이 동기화되지 않음)
- 롤 포워드 (파일에 commit된 데이터와 commit되지 않은 트랜잭션 모두)
- 데이터베이스 open
- 롤백 (파일에 commit되지않은 데이터)

Instance Recovery 튜닝

- Instance Recovery동안 체크포인트 위치와 리두 로그 끝 사이의 트랜잭션이 데이터 파일에 적용되어야 한다.
- 체크포인트 위치와 리두 로그 끝 사이의 차이를 제어함으로써 Instance Recovery 튜닝

Media Recovery 종류

Complete Recovery : recovery가 요청된 시점까지 커밋된 모든 데이터 변경 사항을 포함하여 최신의 상태로 복구

Incomplete Recovery : recovery가 요청되기 전의 지정된 과거 시점으로 복구

12. 백업 및 Recovery 구성

Fast Recovery Area 구성

Fast Recovery Area : Oracle DB의 복구와 관련된 파일들의 통합된 저장 영역

- 간편한 백업 저장 영역 관리를 위해 권장
- 작업 중인 데이터베이스 파일과는 구분되는 저장 공간
- 위치는 DB_RECOVERY_FILE_DEST 파라미터로 지정
- 크기는 DB_RECOVERY_FILE_DEST_SIZE 파라미터로 지정
- Oracle DB의 복구와 관련된 파일들을 모두 보관할 수 있을 만큼 커야 한다.

Control File 다중화

ASM

- 초기화 파라미터 설정 변경
 - ALTER SYSTEM SET control_files =
'+DATA\orcl\controlfile\current.260.836857429' ,
'+DATA' SCOPE = SPFILE ;
- 데이터베이스 instance 종료
 - shutdown immediate
- 데이터베이스 NOMOUNT
 - rman target /
 - startup nomount
- RMAN툴을 사용하여 기존 Control File 복사
 - RESTORE CONTROLFILE FROM
'+DATA\orcl\controlfile\current.260.836857429' ;
- 데이터베이스 MOUNT
 - alter database mount ;
- 데이터베이스 Open
 - alter database open ;

File System

- 초기화 파라미터 설정 변경
 - ALTER SYSTEM SET control_files =
'C:\app\WORK\DISK1\control01.ctl' ,
'C:\app\WORK\DISK2\control02.ctl' SCOPE = SPFILE ;
- 데이터베이스 Instance 종료
 - shutdown immediate
- 운영 체제 명령을 사용하여 기존 Control File 복사
 - host copy C:\app\WORK\oradata\orcl\control01.ctl C:\app\WORK\DISK1\control01.ctl
 - host copy C:\app\WORK\oradata\orcl\control01.ctl C:\app\WORK\DISK2\control02.ctl

- 데이터베이스 Open
 - startup

리두 로그 파일 다중화

- 리두 로그 그룹 정보 확인
 - SELECT group#, sequence#, status, archived, members, bytes/ 1024/ 1204 size_mb
FROM v\$log ;
 - SELECT group#, member, status
FROM v\$logfile
ORDER BY group# ;
- 각 그룹마다 멤버 파일 추가
 - ALTER DATABASE ADD LOGFILE MEMBER
'C:\app\WORK\WORKDISK1\redo01a.log' TO GROUP 1,
'C:\app\WORK\WORKDISK1\redo02a.log' TO GROUP 2,
'C:\app\WORK\WORKDISK1\redo03a.log' TO GROUP 3,
'C:\app\WORK\WORKDISK1\redo01b.log' TO GROUP 1,
'C:\app\WORK\WORKDISK1\redo02b.log' TO GROUP 2,
'C:\app\WORK\WORKDISK1\redo03b.log' TO GROUP 3 ;
- 수동 Log Switch 및 강제 Check Point 수행
 - ALTER SYSTEM SWITCH LOGFILE ;
 - ALTER SYSTEM CHECKPOINT ;

아카이브로그 모드

아카이브로그 모드

- NOARCHIVELOG 모드 : 로그 스위치가 발생할 때마다 리두 로그 파일을 덮어 쓴다.
- ARCHIVELOG 모드 : 로그 스위치가 발생할 때 리두 로그 파일을 지정된 위치에 복사 한다.

ARCn 프로세스

- 선택적 백그라운드 프로세스
- 데이터베이스가 ARCHIVELOG 모드인 경우 활성화
- 데이터베이스에서 수행한 모든 변경 사항을 기록

ASM

ASM(Automatic Storage Management) : 오라클에서 스토리지를 직접 관리하는 방식

- File System과 Raw Device의 장점만을 결합한 방식
 - File System
 - OS를 통하여 데이터를 저장하고 관리하는 방식
 - 사용자 편의성이 좋다.

- OS의존도가 높아 OS의 성능에 따라 오라클의 성능에 영향을 받는다.
- Raw Device
 - 오라클에서 직접 storage에 데이터를 저장하는 방식
 - OS를 거치지 않고 직접 디스크에 I/O를 발생하기 때문에 속도가 빠르다.
 - 관리하기 어렵다.
- ASM 특징
 - 효율적인 디스크 관리 : 디스크 그룹에 새로운 디스크를 추가, 제거하면 ASM에서 자동으로 Rebalancing 작업 수행
 - Rebalancing : 디스크를 추가하거나 제거하면 자동적으로 데이터를 균등하게 분산
 - 디스크 I/O의 효과적 분산 : AU(Allocation Unit) 단위로 나누어서 서로 다른 디스크에 균등하게 분산시켜 저장
 - 비용 절감 : 고가의 RAID 장비나 별도의 하드웨어, 소프트웨어 없이 오라클에서 소프트웨어적으로 관리 가능

13. 데이터베이스 백업 수행

백업의 종류

수동 백업 (User Managed Backup) : 데이터베이스 구동에 필요한 파일을 별도의 저장매체에 직접 백업하는 작업

- 운영조건에 따라 Hot 백업, Cold 백업을 선택하여 수행
 - Hot 백업 : 데이터베이스 Open 상태에서 백업 진행 (백업 중에도 데이터베이스 서비스 가능)
 - 아카이브로그 모드에서만 가능
 - 데이터 파일, 컨트롤 파일만 백업 가능
 - Hot 백업 중에 DML작업으로 변경 된 내용들은 리두 로그 파일에 저장되었다가 백업이 끝나면 데이터파일에 적용
 - Cold 백업 : 데이터베이스를 정상 종료 시킨 후 백업 (백업 중에는 데이터베이스 서비스 불가능)
 - 모든 파일의 시점이 동일해야 한다.
 - 아카이브로그 모드, 노아카이브로그 모드 모두 사용 가능
 - 데이터 파일, 컨트롤 파일, 리두 로그 파일 모두 백업 가능

RMAN 백업 (Recovery Managed Backup) : Recovery Manager라는 오라클 패키지를 이용하여 백업

- Backup set : 여러 데이터 파일에서 실제로 사용되는 블록만 추출하여 하나의 백업 파일로 묶어서 저장
 - 저장 공간 절약
 - 데이터 파일 복구 시 Restore 작업이 필요하므로 작업 시간이 길다.
 - Full 백업, Incremental 백업 가능
- Image copy : 데이터 파일별로 각각의 백업 파일을 만들어서 저장 (빈 블록을 포함)
 - 많은 저장 공간 소요
 - 데이터 파일 복구 시 Restore 작업이 필요 없기 때문에 작업 시간이 짧다.
 - Full 백업만 가능
 - Full 백업 : 사용된 모든 데이터 파일을 복사하는 작업
 - Incremental 백업
 - cumulative 백업
 - differential 백업

백업 솔루션을 이용한 백업 (Third Party Solution)

- 위의 두기 백업 방법을 이용하여 편하도록 Third Party Solution을 만든 프로그램
 - Netbackup
 - Backup Xcelerator

Control File을 Trace File로 백업

Trace File : 데이터베이스에 문제가 발생했을 때 그 문제를 진단하고 디버깅할 수 있는 정보를 제공하는 파일

- Control File을 재생성하는 SQL 명령을 통해 Control File을 Trace File로 백업
- 모든 Control File이 손실될 경우 Control File Trace 백업을 사용하여 복구 가능

14. 데이터베이스 Recovery 수행

Control File 복구

ASM

- Enterprise Manager를 사용하여 단계별 Recovery 수행
- RMAN을 사용하여 Recovery 수행
 - 데이터베이스 NOMOUNT모드로 설정
 - RMAN에 연결하고 RESTORE CONTROLFILE 명령을 실행하여 Control File 복원
 - Control File 복원 후 데이터베이스 Open

파일 시스템

- 데이터베이스 종료
- 손실된 Control File을 대체할 다중화 된 Control File을 복사

리두 로그 파일 복구

리두 로그 파일 손실

- 정상적인 Instance 작업에는 영향을 주지 않는다.
- Alert log를 통해 누락된 로그 파일을 확인한다.

리두 로그 파일 복구

- 손실된 리두 로그 멤버 삭제
- 새 멤버를 추가

데이터 파일 복구

NOARCHIVELOG 모드

- 데이터베이스 종료
- 모든 데이터 파일 및 Control File을 포함하여 전체 데이터베이스를 백업에서 복원
- 데이터베이스 Open
- 마지막 백업 이후 변경된 모든 사항 재입력

ARCHIVELOG 모드

- Noncritical 데이터 파일 손실 (SYSTEM 또는 UNDO 테이블스페이스에 속하지 않는 데이터 파일 손상/손실)
 - 누락된 데이터 파일을 복원하고 복구

- Critical 데이터 파일 손실 (SYSTEM 또는 UNDO 테이블스페이스에 속하는 데이터 파일 손상/손실)
 - 데이터베이스 종료
 - 데이터베이스 MOUNT모드로 설정
 - 누락된 데이터 파일을 복원하고 복구
 - 데이터베이스 Open

15. 데이터 이동

Oracle Data Pump

Data Pump : 고속 데이터/메타 데이터 이동을 위한 서버 기반 기능

- DBMS_DATAPUMP 패키지를 통해 호출
- 다양한 데이터 이동 방법 제공
 - o 데이터 파일 복사
 - o Direct path
 - o External Table
 - o 네트워크 링크 지원

Data Pump 이점

- JOB 컨트롤 기능 : Interactive mode를 통해 Data Pump 작업 통제 가능
 - Data Pump 작업을 중단/재시작 가능
 - 동적으로 Dump File 할당 가능
 - 장애가 발생하여 작업이 중지되어도 장애 복구 후 재시작 가능
- 병렬 수행 지원 : PARALLEL 파라미터를 통해 Data Pump 작업의 프로세스를 병렬화 가능
 - 여러개의 데이터 파일에 데이터를 쓰거나 여러개의 데이터 파일로부터 데이터를 읽어 데이터베이스에 저장
 - 병렬 수행을 지원함으로써 수행 속도 단축
- 작업에 필요한 공간 예상 : ESTIMATE_ONLY 파라미터를 통해 작업이 소비하는 공간의 양 예측 가능
- 원격지 수행 : DB LINK를 통해 원격지 데이터에 대한 Data Pump 수행 가능
- Remapping 지원 : 유저 스키마, 테이블 스페이스, 데이터 파일 등과 같은 정보들을 Data Pump시 변경 가능

SQL Loader

SQL Loader : 외부 데이터 파일의 데이터를 데이터베이스의 테이블로 넣기 위한 유틸리티

- Data File : SQL Loader를 사용하여 데이터베이스에 입력될 텍스트 형식으로 구성된 데이터 파일
 - 제어 파일에 지정된 하나 이상의 파일에서 데이터를 읽는다.
 - 데이터 파일의 데이터는 레코드로 구성
- Control File : SQL Loader를 사용하는데 필수적인 파일로써 데이터 정의어 지침을 포함하는 텍스트 파일
 - 데이터 검색 위치
 - 데이터 구문 분석 및 해석 방법
 - 데이터 삽입 위치
- Log File : 데이터 로드 상태 정보, 로드 된 행의 수, 로드 진행중 거부된 행의 수 등 자세한 테이블 로드 정보 포함

- Bad File : SQL Loader 또는 오라클 데이터베이스가 거부한 레코드를 포함
 - 입력 형식에 부적합할 경우
 - 제약조건을 위해 하였을 경우
- Discard File : SQL Loader가 실행되는 동안 폐기되는 파일
 - 컨트롤 파일에 명시한 WHEN절의 조건에 맞지 않는 데이터를 저장

로드 방식

Conventional Path Load : SQL Loader에서 설정된 일정량의 버퍼를 채우면 SQL Insert 문장을 이용해 로딩

- Conventional Path Load 방식을 사용해야 하는 상황
 - 데이터 로딩 중에도 DML이 계속 수행돼야 하는 경우
 - Direct Path 방식을 지원하지 않는 클러스터 테이블을 로딩해야 하는 경우
 - 데이터 로딩 중에 제약조건에 의해 에러가 발생한 데이터를 걸러야 할 경우
 - 사이즈가 큰 인덱스가 있는 테이블에 적은 양의 데이터를 로딩해야 할 경우
 - 데이터 로딩 중에 트리거가 동작해야 할 경우

Direct Path Load : 버퍼 캐시를 거치지 않고 디스크에 직접 로딩

	Conventional Path	Direct Path
속도	느림	빠름
가용성	로드 작업 중 다른 사용자가 테이블 수정 가능	로드 작업 중 다른 사용자가 테이블 수정 불가
리두 생성 조건	항상	Archive mode에서만 리두 생성
클러스터 테이블	클러스터 테이블에 load 가능	클러스터 테이블에 load 불가
제약 조건	모든 제약 조건 검증	PRIMARY KEY, UNIQUE, NOT NULL만 검증
INSERY 트리거	실행	실행되지 않음

External Table

External Table : 오라클 데이터베이스 외부의 운영체제에 파일로 저장되는 읽기 전용 테이블

External Table 이점

- 외부의 파일에서 직접 데이터를 사용하거나 다른 데이터베이스로 데이터를 로드할 수 있다.

- 외부의 데이터를 로드할 필요 없이 데이터베이스에 상주하는 테이블과 병렬로 직접 Join 및 Query할 수 있다.
- 복합 Query의 결과는 외부의 파일로 언로드할 수 있다.
- 로드를 위해 다른 소스에서 생성된 파일을 결합할 수 있다.

16. 데이터베이스 유지 관리

AWR(Automatic Workload Repository)

AWR(Automatic Workload Repository) : 데이터베이스 서버에서 정기적으로 데이터베이스에 대한 통계 및 성능자료 등을 수집해 스냅샷을 만들고 그 데이터를 저장하는 공간

- AWR에 수집된 정보를 통해 서버 튜닝, 데이터베이스 관리, 장애와 복구에 대한 효과적인 작업 가능
- 데이터베이스 SGA에서 60분마다 자동으로 통계 정보를 캡처하여 스냅샷의 형태로 AWR에 저장
- 스냅샷은 MMON 백그라운드 프로세스에 의해 디스크에 저장
 - MMON(Memory Monitor) : 자가 진단을 위해 메모리에 존재하는 데이터베이스 통계 정보를 주기적으로 AWR에 저장
- 스냅샷은 기본적으로 8일간 보관
- Repository 테이블에 대한 DML 명령은 지원되지 않음

AWR 관리

- statistics_level : 자동 유지 관리 작업을 포함하여 다양한 통계 및 Advisor캡처를 제어하는 파라미터
 - BASIC : AWR 통계 및 metrics가 계산되지 않고 ASMM을 사용할 수 없으며 advisor 기능이 비활성화
 - TYPICAL : 데이터베이스 자체 관리에 필요한 주요 통계 수집
 - 데이터베이스 동작을 모니터링하는데 필요한 항목
 - 오래되거나 잘못된 통계로 인해 SQL문의 성능이 저하될 가능성을 줄여주는 자동 통계 수집 포함
 - ALL : 가능한 모든 통계 캡처
 - 특정 진단 테스트가 필요한 경우 외에는 사용하지 않는 것을 권장

AWR Baseline

AWR Baseline : AWR에서 태그를 지정하고 보존하는 중요 기간에 대한 스냅샷 데이터의 집합

- 성능 저하 문제 발생 시 이 데이터를 기준으로 성능 튜닝을 수행

ADDM(Automatic Database Diagnostic Monitor)

ADDM : AWR 데이터를 이용해 데이터베이스의 문제점을 자동으로 분석하는 기능

- AWR 스냅샷 생성 후 자동으로 실행되며 마지막 두 스냅샷에 해당하는 기간을 분석
- Instance를 사전에 모니터링하여 심각한 문제로 발전하기 전에 대부분의 병목 지점을 감지
- ADDM의 분석 결과는 AWR에 저장

AutoTask(Automated Maintenance Task)

Autotask 기능

- Automatic Optimizer Statics Collection
- Automatic Segment Advisor
- Automatic SQL Tuning Advisor

Autotask 작업 순서

- Maintenance Window가 열린다.
- Autotask 백그라운드 프로세스가 작업 일정을 잡는다.
- Scheduler가 작업을 시작한다.
- Resource Manager가 Autotask작업의 영향을 제한하여 과도한 자원 낭비를 막는다.

17. 성능 관리

성능 관리

성능 계획 : 하드웨어, 소프트웨어, 운영 체제, 네트워크 Infrastructure 등과 같은 환경을 설정하는 과정

- 투자 옵션
- 시스템 구조
- 확장성
- 응용 프로그램 설계 원칙
- 작업 로드 테스트, 모델링 및 구현
- 새 응용 프로그램 배치

Instance Tuning : 데이터베이스의 성능이 향상되도록 데이터베이스 파라미터와 OS 파라미터를 조정

- 명확한 목표 수립
- 데이터베이스 구조에 메모리 할당
- 데이터베이스의 각 부분에서 I/O 요구 사항을 고려
- 데이터베이스가 최적의 성능으로 실행되도록 운영 체제를 튜닝

SQL Tuning : 응용 프로그램이 효율적인 SQL문을 실행하도록 조정

- 잘못 튜닝된 SQL문을 식별
- 개별 명령문을 튜닝
- 전체 응용 프로그램을 튜닝

메모리 관리

메모리 구성 요소 관리

- 수동 관리
- ASMM(Automatic Shared Memory Management) : SGA 구성요소(Shared_pool, DB_buffer_cache, Large_pool, Java_pool, Streams_pool)에 관하여 오라클이 스스로 현재 부하상황을 판단하여 메모리를 관리하는 방법
 - MMAN(Memory Manager) 백그라운드 프로세스가 주기적으로 수집한 작업 부하 정보를 바탕으로 동적으로 구성
 - ASMM 사용 조건
 - STATISTICS_LEVEL 파라미터 값이 TYPICAL 또는 ALL로 설정
 - SGA_TARGET 파라미터 값을 0보다 큰 값으로 설정
 - SGA_TARGET은 SGA_MAX_SIZE보다 반드시 작거나 동일한 값으로 설정

- AMM(Automatic Memory Management) : SGA + PGA를 자동으로 관리하는 방법
 - AMM 사용 조건
 - SGA_TARGET과 PGA_AGGREGATE_TARGET의 값을 0으로 지정
 - MEMORY_TARGET 파라미터 값을 0보다 큰 값으로 설정
 - MEMORY_TARGET은 MEMORY_MAX_TARGET보다 반드시 작거나 동일한 값으로 설정

옵티마이저

옵티마이저 : 옵티마이저 통계를 통해 가장 효율적인 방법으로 SQL을 수행할 최적의 실행 계획을 생성하는 알고리즘

- 표현식 및 조건 평가
- 객체 및 시스템 통계를 사용
- 데이터 access 방법을 결정
- 테이블 조인 방법을 결정
- 가장 효율적인 경로 결정

옵티마이저 통계 수집

- 객체 통계
- 운영 체제 통계

통계 수집 방법

- 자동 : 자동 유지 관리 작업
- 수동 : DBMS_STATS 패키지
- 데이터베이스 초기화 파라미터 설정
- 다른 데이터베이스에서 통계 импорт

SQL Plan Directive

SQL Plan Directive : 옵티마이저가 더 나은 Plan을 생성하는데 사용할 수 있는 추가 정보 및 지침

- 누락된 통계 수집
- 열 그룹 통계 생성
- 동적 샘플링 수행

SQL Plan Directive 유지 및 관리

- SYSAUX 테이블스페이스에 저장
- 자동으로 유지 관리되며, 필요에 따라 생성
- 사용하지 않으면 1년 후 삭제

Adaptive Plan

Adaptive Plan : SQL 실행 중 옵티마이저 예측의 부정확함이 발견되면 Query계획이 변경

- Query 실행 중 수집되는 통계를 기반
- OPTIMIZER_ADAPTIVE_REPORTING_ONLY가 기본값인 FALSE로 설정된 경우 Adaptive 실행 계획 사용

18. Database Resource Manager 사용

Data Resource Manager

Data Resource Manager : 비효율적인 OS 관리로 초래된 문제들을 해결하기 위해 데이터베이스 서버가 자원 관리를 결정

- Resource Manager 구성 요소
 - Resource Consumer Group : 시스템 및 데이터베이스 리소스의 사용이 유사한 사용자 또는 세션의 집합
 - Resource Plan : 다양한 Resource Consumer Group간 리소스 분배 방법을 지정
 - Resource Plan Directive : Consumer Group 또는 Subplan간 특정 리소스 공유 방법을 지정
 - Subplan : Database Resource Manager에 의해 플랜 내부에 플랜을 생성
- Resource Manager 관리 요소
 - CPU : Consumer Group과 Subplan간 CPU 리소스가 할당되는 방식을 지정
 - MGMT_MTH 파라미터를 사용하여 CPU 사용량 할당
 - EMPHASIS : 단일 레벨 또는 다중 레벨 Plan에 사용되며 백분율을 사용하여 CPU가 분배되는 방식을 지정
 - RATIO : 단일 레벨 Plan에 사용되며 비율을 사용하여 CPU가 분배되는 방식을 지정
 - Degree of Parallelism Limit : Consumer Group내의 모든 작업에 대하여 최대 병렬 처리 수준을 제어
 - Active Session Pool with Queuing : Consumer Group 또는 Subplan에 대한 동시 활성 세션 수를 제한
 - DBA가 모든 Consumer Group에 사용되는 자원의 양을 간접적으로 제어 가능
 - 시스템 내의 자원을 점유하는 서버의 개수를 감소 (비효율적인 페이징, 스와핑 및 리소스 부족 문제 해결)
 - Active Session Pool이 활성화 세션으로 가득 차면 다른 활성화 세션이 완료되거나 비활성화 될 때까지 활성화를 시도하는 모든 후속 세션을 큐에 저장
 - Undo Pool : Consumer Group 또는 Subplan에 의해 생성될 수 있는 총 언두 크기를 제어
 - Execution Time Limit : 작업에 허용되는 최대 실행 시간을 지정
 - Idle Time Limit : 세션이 Idle 상태로 있는 시간을 지정
 - 세션이 지정된 제한 시간을 초과하면 PMON 프로세스가 강제로 세션을 종료하고 해당 세션의 상태를 정리
 - Consumer Group Switching : 세션에서 Consumer Group을 전환하게 하는 조건을 지정
 - Database Consolidation : Resource Manager를 사용하여 동시 데이터베이스 세션 간에 리소스 할당 최적화
 - Database Consolidation을 수행하려면 응용 프로그램들을 서로 분리해야 한다.
 - 각 응용 프로그램끼리 서로 영향을 미치면 안된다.
 - 각 응용 프로그램의 성능에는 일관성이 있어야 한다.
 - Server Consolidation : 대부분의 테스트, 개발 및 소규모 운용 데이터베이스는 서버를 완전히 활용하지 못하므로 Server Consolidation을 사용
 - CPU 경합이 발생 할 수 있으며 하나의 Instance에 작업 로드 과부하로 인해 악영향을 끼칠 수 있다.
 - Instance Caging : CPU_COUNT 초기화 파라미터를 사용하여 하나의 Instance가 사용할 수 있는 CPU 수를 제한
 - Over Provisioning 방식 : non-critical 데이터베이스 및 로드가 낮은 non-critical 운용 시스템에 적합
 - 각 Instance에 대한 CPU 제한의 합이 실제 CPU 수를 초과
 - 각 Instance가 서로의 성능에 영향을 준다.

- Partitioning 방식 : critical 제품 시스템에 적합
 - 각 Instance에 대한 CPU 제한의 합이 실제 CPU 수와 동일
 - 각 Instance의 CPU 리소스가 고정되므로 서로 다른 Instance에 영향을 줄 수 없다.

19. Oracle Scheduler 사용

Scheduler

Scheduler : 데이터베이스 관리자와 응용 프로그램 개발자가 Scheduler를 사용하여 데이터베이스 환경에서 여러 작업이 수행되는 시기와 위치를 제어

Scheduler 구성 요소

- 작업(Job) : 프로그램과 스케줄을 지정
- 스케줄(Schedule) : 작업의 실행 시기와 횟수를 지정
 - Time-Based
 - Calendaring 표현식 : 반복 간격과 작업 시작 날짜를 사용하여 다음 작업의 시작 시간 계산
 - INCLUDE : Calendaring 표현식 결과에 날짜 리스트 추가
 - EXCLUDE : Calendaring 표현식 결과에 날짜 리스트 제거
 - INTERSECT : 두 개 이상의 스케줄에 공통되는 날짜만 사용
 - Date-Time 표현식 : 지정된 표현식에 따라 다음 작업 실행 시간 결정
 - Event-Based : 특정 이벤트 발생시 작업 시작
- 프로그램(Program) : 수행 프로그램 및 argument 지정

작업 체인

체인 : 결속된 목표를 위해 함께 연결된 일련의 명명된 프로그램 (종속형 스케줄링)

작업 체인 생성

- 체인 객체 생성 : CREATE_CHAIN 프로시저를 사용하여 체인을 생성
- 체인 단계 정의 : 단계 정의 시 이름을 부여하고 단계 중 발생하는 사항을 지정
- 체인 규칙 정의 : 단계가 실행되는 시기와 단계 간의 종속성을 정의
- 체인 시작
 - ENABLE 프로시저를 사용하여 체인을 활성화
 - CHAIN 유형의 작업을 생성

고급 Scheduler

고급 Scheduler 기능을 사용하여 작업 윈도우 및 작업 우선 순위 지정과 같은 스케줄링의 다양한 측면을 더 세밀하게 제어

- 윈도우 : 시작 시간과 종료 시간이 시간 간격으로 표시

- 시간대 별로 다양한 Resource Plan을 활성화하는데 사용
- 윈도우 그룹 : 윈도우 리스트
 - 윈도우를 보다 쉽게 관리
- 작업 클래스 : 공통적인 리소스 사용 요구사항과 기타 특성을 공유하는 작업 카테고리를 정의
- Resource Consumer Group : 작업 클래스의 작업에 할당되는 리소스를 결정
- Resource Plan : Resource Consumer Group에서 리소스의 우선 순위를 지정