

## **Oracle Database 11g: Administration Workshop I(한글판)**

**볼륨 I • 학생용**

D50102KR20

Edition 2.0

2010년 5월

D67248

**ORACLE®**

## 만든이

Deirdre Matishak

Mark Fuller

## 기술 제공자 및 검토자

Maria Billings

Herbert Bradbury

Yanti Chang

Timothy Chien

Andy Fotunak

Gerlinde Frenzen

Steve Friedberg

Joel Goodman

Vimala Jacob

Dominique Jeunot

Pete Jones

Fukue Kawabe

Donna Keesling

Sean Kim

Achiel Langers

Gwen Lazenby

Essi Parast

Randy Richeson

Joe Roch

Hilda Simon

Ira Singer

Jim Spiller

Supithran Thananayagam

Branislav Valny

Manju Varrier

## 편집자

Raj Kumar

Daniel Milne

## 그래픽 디자이너

Rajiv Chandrabhanu

## 발행인

Jobi Varghese

Veena Narasimhan

**Copyright © 2010, Oracle. All rights reserved.**

### Disclaimer

본 문서는 독점적 정보를 포함하고 있으며 저작권법 및 기타 지적 재산법에 의해 보호됩니다. 본 문서는 오라클 교육 과정에서 자신이 사용할 목적으로만 복사하고 인쇄할 수 있습니다. 어떤 방법으로도 본 문서를 수정하거나 변경할 수 없습니다. 저작권법에 따라 "공정"하게 사용하는 경우를 제외하고, 오라클의 명시적 허가 없이 본 문서의 전체 또는 일부를 사용, 공유, 다운로드, 업로드, 복사, 인쇄, 표시, 실행, 재생산, 게시, 라이센스, 우편 발송, 전송 또는 배포할 수 없습니다.

본 문서의 내용은 사전 공지 없이 변경될 수 있습니다. 만일 본 문서의 내용상 문제점을 발견하면 서면으로 통지해 주기 바랍니다. Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. 오라클은 본 문서에 오류가 존재하지 않음을 보증하지 않습니다.

### Restricted Rights Notice

만일 본 문서를 미국 정부나 또는 미국 정부를 대신하여 문서를 사용하는 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

### U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

### Trademark Notice

Oracle은 Oracle Corporation 또는 그 자회사의 등록 상표입니다.  
기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

# 목차

## I 소개

- 과정 목표 I-2
- 제안 일정 I-3
- 오라클 제품 및 서비스 I-4
- Oracle Database 11g: "g"는 그리드(Grid)의 약어 I-5
- 단일 Instance 용 그리드 Infrastructure I-7

## 1 오라클 데이터베이스 구조 탐색

- 목표 1-2
- 오라클 데이터베이스 1-3
- 서버에 연결 1-4
- 오라클 데이터베이스 서버 구조: 개요 1-6
- Instance: 데이터베이스 구성 1-7
- 데이터베이스 Instance에 연결 1-8
- 오라클 데이터베이스 메모리 구조 1-9
- Shared Pool 1-11
- 데이터베이스 버퍼 캐시 1-13
- 리두 로그 버퍼 1-14
- Large Pool 1-15
- Java Pool 및 Streams Pool 1-16
- PGA(Program Global Area) 1-17
- 퀴즈 1-18
- 프로세스 구조 1-20
- DBW $n$ (데이터베이스 기록자 프로세스) 1-23
- LGWR(로그 기록자 프로세스) 1-25
- CKPT(체크포인트 프로세스) 1-27
- SMON(시스템 모니터 프로세스) 1-28
- PMON(프로세스 모니터 프로세스) 1-29
- 복구자 프로세스 1-30
- ARC $n$ (아카이버 프로세스) 1-31
- 프로세스 시작 시퀀스 1-32
- 데이터베이스 저장 영역 구조 1-33
- 논리적 및 물리적 데이터베이스 구조 1-35
- 세그먼트, Extent 및 블록 1-37

테이블스페이스 및 데이터 파일 1-38  
SYSTEM 및 SYSAUX 테이블스페이스 1-39  
자동 저장 영역 관리(ASM) 1-40  
ASM 저장 영역 구성 요소 1-41  
오라클 데이터베이스와 상호 작용: 메모리, 프로세스 및 저장 영역 1-42  
퀴즈 1-44  
요약 1-46  
연습 1: 개요 1-47

## 2 Oracle 소프트웨어 설치

목표 2-2

오라클 데이터베이스 관리자의 작업 2-3  
오라클 데이터베이스 관리 도구 2-4  
설치 계획 2-6  
Oracle 그리드 Infrastructure 및 오라클 데이터베이스 설치: 시스템  
요구 사항 2-8  
운영 체제 준비 2-9  
환경 변수 설정 2-10  
시스템 요구 사항 검사 2-11  
OUI(Oracle Universal Installer) 2-12  
예제: 설치 시나리오 2-13  
1부: 독립형 서버를 위한 Oracle 그리드 Infrastructure 설치 2-14  
제품 언어 선택 2-15  
ASM 디스크 그룹 생성 2-16  
ASM 암호 정의 2-17  
권한이 있는 운영 체제 그룹 정의 2-18  
설치 위치 지정 2-19  
Inventory 생성 2-20  
필수 구성 요소 검사 수행 2-21  
설치 요약 데이터 확인 2-22  
설치 진행 상황 모니터 2-23  
root 구성 스크립트 실행 2-24  
Configuration Assistants 실행 2-25  
설치 완료 2-26  
FRA 디스크 그룹 구성 2-27  
퀴즈 2-28  
2부: 오라클 데이터베이스 소프트웨어 설치 2-30  
설치 유형 선택 2-31  
그리드 설치 옵션 선택 2-32

언어 설정 선택	2-33
데이터베이스 버전 선택	2-34
설치 위치 지정	2-35
운영 체제 그룹 선택	2-36
필수 구성 요소 검사 수행	2-37
설치 요약 페이지	2-38
제품 설치 페이지	2-39
설치 완료	2-40
설치 옵션: 자동 모드	2-41
퀴즈	2-42
요약	2-44
연습 2 개요: 데이터베이스 환경 준비	2-45

### 3 DBCA 를 사용하여 오라클 데이터베이스 생성

목표	3-2
데이터베이스 계획	3-3
데이터베이스: 예제	3-4
적합한 Character Set 선택	3-5
Character Set 사용 방법	3-7
방지해야 할 문제	3-8
DBCA(Database Configuration Assistant)	3-9
DBCA 를 사용하여 데이터베이스 생성	3-10
데이터베이스 요약 생성	3-16
암호 관리	3-17
데이터베이스 설계 템플릿 생성	3-18
DBCA 를 사용하여 데이터베이스 삭제	3-19
추가 작업에 DBCA 사용	3-21
퀴즈	3-22
요약	3-24
연습 3 개요: DBCA 사용	3-25

## 4 데이터베이스 Instance 관리

목표 4-2

관리 프레임워크 4-3

Database Control 시작 및 정지 4-4

Oracle Enterprise Manager 4-5

Database Home 페이지 4-7

다른 오라클 도구 4-8

SQL\*Plus 사용 4-9

셸 스크립트에서 SQL\*Plus 호출 4-10

SQL\*Plus에서 SQL 스크립트 호출 4-11

초기화 파라미터 파일 4-12

간소화된 초기화 파라미터 4-14

초기화 파라미터: 예제 4-15

SQL\*Plus를 사용하여 파라미터 확인 4-19

초기화 파라미터 값 변경 4-21

파라미터 값 변경: 예제 4-23

퀴즈 4-24

데이터베이스 시작 및 종료: 인증서 4-26

오라클 데이터베이스 Instance 시작 4-27

오라클 데이터베이스 Instance 시작: NOMOUNT 4-28

오라클 데이터베이스 Instance 시작: MOUNT 4-29

오라클 데이터베이스 Instance 시작: OPEN 4-30

시작 옵션: 예제 4-31

오라클 데이터베이스 Instance 종료 4-32

종료 모드 4-33

종료 옵션 4-34

종료 옵션: 예제 4-37

Alert Log 확인 4-38

Trace file 사용 4-40

Dynamic Performance 뷰 4-42

Dynamic Performance 뷰: 사용 예제 4-43

Dynamic Performance 뷰: 고려 사항 4-44

데이터 딕셔너리: 개요 4-45

데이터 딕셔너리 뷰 4-46

데이터 딕셔너리: 사용 예제 4-48

퀴즈 4-49

요약 4-51

연습 4 개요: Oracle Instance 관리 4-52

## 5 ASM Instance 관리

목표 5-2

관리자를 위한 ASM 이점 5-3

ASM Instance 5-4

ASM 구성 요소: ASM Instance—기본 프로세스 5-6

ASM Instance 초기화 파라미터 5-7

데이터베이스 Instance 와 ASM 사이의 상호 작용 5-9

ASM Instance: Dynamic Performance 뷰 5-10

ASM 시스템 권한 5-11

Enterprise Manager 를 사용하여 ASM 유저 관리 5-12

SQL\*Plus 를 사용하여 ASM Instance 시작 및 정지 5-13

srvctl 을 사용하여 ASM Instance 시작 및 정지 5-15

asmcmd 를 사용하여 ASM Instance 시작 및 정지 5-16

디스크 그룹 개요 5-17

ASM 디스크 5-18

할당 단위(AU: Allocation Unit) 5-19

ASM 파일 5-20

Extent 맵 5-21

스트라이핑 세분성 5-22

하위단계(Fine-Grain) 스트라이핑 5-23

ASM Failure 그룹 5-25

스트라이핑 및 Mirroring 예제 5-26

Failure 예제 5-27

디스크 그룹 관리 5-28

SQL\*Plus 를 사용하여 디스크 그룹 생성 및 삭제 5-29

디스크 그룹에 디스크 추가 5-30

기타 ALTER 명령 5-31

Enterprise Manager 를 사용한 ASM 관리 5-32

ASM 디스크 그룹 호환성 5-33

ASM 디스크 그룹 속성 5-35

Enterprise Manager 를 사용하여 디스크 그룹 속성 편집 5-36

ASM 메타 데이터 검색 5-37

ASM Fast Mirror Resync 개요 5-38

퀴즈 5-39

요약 5-41

연습 5 개요: ASM Instance 관리 5-42

## 6 Oracle 네트워크 환경 구성

목표 6-2

Oracle Net 서비스 6-3

Oracle Net 리스너 6-4

Net 연결 설정 6-5

연결 설정 6-6

유저 세션 6-7

Oracle 네트워크 구성 및 관리 도구 6-8

Listener Control 유틸리티 6-10

Listener Control 유틸리티 구문 6-11

SRVCTL 을 사용하여 리스너 시작 및 정지 6-13

Listener Home 페이지 6-14

Net Services Administration 페이지 6-15

리스너 생성 6-16

리스너 주소 추가 6-17

데이터베이스 서비스 등록 6-18

이름 지정 방식 6-20

간단한 연결(Easy Connect) 6-21

로컬 이름 지정 6-22

디렉토리 이름 지정 6-23

외부 이름 지정 방식 6-24

서비스 Alias 구성 6-25

고급 연결 옵션 6-26

Oracle Net 연결 테스트 6-28

유저 세션: Dedicated Server 프로세스 6-29

유저 세션: Shared Server 프로세스 6-30

SGA 및 PGA 6-31

Shared Server: 연결 풀링 6-32

Shared Server 를 사용하지 않는 경우 6-33

데이터베이스 간의 통신 구성 6-34

다른 데이터베이스에 연결 6-35

퀴즈 6-36

요약 6-38

연습 6 개요: Oracle 네트워크 구성 요소 사용 6-39

## 7 데이터베이스 저장 영역 구조 관리

### 목표 7-2

- 테이블 데이터가 저장되는 방법 7-3
- 데이터베이스 블록: 내용 7-4
- 저장 영역 구조 탐색 7-5
- 새 테이블스페이스 생성 7-6
- 테이블스페이스에 대한 저장 영역 7-8
- 미리 구성된 데이터베이스의 테이블스페이스 7-10
- 테이블스페이스 변경 7-12
- 테이블스페이스에서 가능한 작업 7-14
- 테이블스페이스 삭제 7-16
- 테이블스페이스 정보 보기 7-17
- 테이블스페이스 내용 보기 7-18
- OMF(Oracle 관리 파일) 7-19
- 데이터베이스 확장 7-21
- 퀴즈 7-22
- 요약 7-24
- 연습 7 개요: 데이터베이스 저장 영역 구조 관리 7-25

## 8 유저 보안 관리

### 목표 8-2

- 데이터베이스 유저 계정 8-3
- 미리 정의된 관리 계정 8-5
- 유저 생성 8-6
- 유저 인증 8-7
- 관리자 인증 8-9
- 유저 계정 Lock 해제 및 암호 재설정 8-10
- 권한 8-11
- 시스템 권한 8-12
- 객체 권한 8-14
- ADMIN OPTION 을 사용하여 시스템 권한 취소 8-15
- GRANT OPTION 을 사용하여 객체 권한 취소 8-16
- 롤 사용 시의 이점 8-17
- 롤에 권한 할당 및 유저에게 룰 할당 8-18
- 미리 정의된 룰 8-19
- 룰 생성 8-20
- 보안 룰 8-21
- 유저에게 룰 할당 8-22
- 퀴즈 8-23

프로파일 및 유저	8-25
암호 보안 기능 구현	8-27
암호 프로파일 생성	8-29
제공된 암호 확인 함수: VERIFY_FUNCTION_11G	8-30
유저에게 할당량 할당	8-31
최소 권한의 원칙 적용	8-33
권한이 부여된 계정 보호	8-35
퀴즈	8-36
요약	8-38
연습 8 개요: 유저 관리	8-39

## 9 데이터 동시성 관리

목표	9-2
Lock	9-3
Lock 메커니즘	9-4
데이터 동시성	9-5
DML Lock	9-7
Enqueue 메커니즘	9-8
Lock 충돌	9-9
Lock 충돌 유발 원인	9-10
Lock 충돌 감지	9-11
Lock 충돌 해결	9-12
SQL을 사용하여 Lock 충돌 해결	9-13
Deadlock	9-14
퀴즈	9-15
요약	9-17
연습 9 개요: 데이터 및 동시성 관리	9-18

## 10 언두 데이터 관리

목표	10-2
언두 데이터	10-3
트랜잭션 및 언두 데이터	10-5
언두 정보 저장	10-6
언두 데이터와 리두 데이터 비교	10-7
언두 관리	10-8
언두 Retention 구성	10-9
언두 Retention 보장	10-11

언두 테이블스페이스를 고정 크기로 변경	10-12
일반 언두 정보	10-13
Undo Advisor 사용	10-14
시스템 작업 보기	10-15
퀴즈	10-16
요약	10-18
연습 10 개요: 언두 세그먼트 관리	10-19

## 11 오라클 데이터베이스 감사(audit) 구현

목표	11-2
책임 구분	11-3
데이터베이스 보안	11-4
준수 여부 모니터	11-6
표준 데이터베이스 감사(Audit)	11-7
Audit Trail 구성	11-8
Audit Trail 균일화	11-9
감사(Audit) 옵션 지정	11-10
기본 감사(Audit)	11-11
Enterprise Manager Audit 페이지	11-12
감사(Audit) 정보 사용 및 유지 관리	11-13
값 기준 감사(Audit)	11-14
Fine-Grained Auditing	11-16
FGA 정책	11-17
감사(Audit)된 DML 문: 고려 사항	11-19
FGA 지침	11-20
SYSDBA 감사(Audit)	11-21
Audit Trail 유지 관리	11-22
Oracle Audit Vault	11-23
퀴즈	11-24
요약	11-26
연습 11 개요: 오라클 데이터베이스 보안 구현	11-27

## 12 데이터베이스 유지 관리

목표 12-2

데이터베이스 유지 관리 12-3

Alert History 확인 12-4

용어 12-5

Oracle 옵티마이저: 개요 12-6

옵티마이저 통계 12-7

Manage Optimizer Statistics 페이지 사용 12-8

수동으로 옵티마이저 통계 수집 12-9

통계 수집 환경 설정 12-11

AWR(Automatic Workload Repository) 12-13

AWR Infrastructure 12-14

AWR Baseline 12-15

Enterprise Manager 및 AWR 12-16

AWR 관리 12-17

통계 레벨 12-18

ADDM(Automatic Database Diagnostic Monitor) 12-19

ADDM 결과 12-20

ADDM 권장 사항 12-21

Advisory 프레임워크 12-22

Enterprise Manager 및 Advisor 12-24

DBMS\_ADVISOR 패키지 12-25

퀴즈 12-26

자동화된 유지 관리 업무 12-27

자동화된 유지 관리 업무 구성 12-29

서버 생성 Alert 12-30

임계값 설정 12-31

Alert 생성 및 테스트 12-32

Alert 통지 12-33

Alert에 대한 대응 12-35

Alert 유형 및 Alert 지우기 12-36

퀴즈 12-37

요약 12-38

연습 12 개요: Proactive Maintenance 12-39

## 13 성능 관리

- 목표 13-2
- 성능 모니터 13-3
- Enterprise Manager Performance 페이지 13-4
- 특정 대기 범주로 드릴 다운 13-5
- Performance 페이지: 처리 능력 13-6
- 성능 모니터: Top Sessions 13-7
- 성능 모니터: Top Services 13-8
- 메모리 구성 요소 관리 13-9
- AMM(자동 메모리 관리) 활성화 13-10
- ASMM(자동 공유 메모리 관리) 활성화 13-11
- Automatic Shared Memory Advisor 13-12
- Dynamic Performance 통계 13-13
- 문제 해결 및 튜닝 뷰 13-15
- 부적합한 객체 및 사용할 수 없는 객체 13-16
- 퀴즈 13-18
- 요약 13-20
- 연습 13 개요: 성능 모니터 및 개선 13-21

## 14 백업 및 Recovery 개념

- 목표 14-2
- 관리자의 임무 14-3
- Failure의 범주 14-5
- Statement Failure 14-6
- User Process Failure 14-7
- Network Failure 14-8
- User Error 14-9
- Flashback 기술 14-10
- Instance Failure 14-12
- Instance Recovery 이해: CKPT(체크포인트) 프로세스 14-13
- Instance Recovery 이해: 리두 로그 파일 및 로그 기록자 14-14
- Instance Recovery 이해 14-15
- Instance Recovery의 단계 14-16
- Instance Recovery 튜닝 14-17
- MTTR Advisor 사용 14-18
- Media Failure 14-19
- Recovery 가능성 구성 14-20
- Fast Recovery Area 구성 14-21
- 콘트롤 파일 다중화 14-22
- 리두 로그 파일 14-24

리두 로그 다중화	14-25
아카이브 로그 파일	14-26
ARCn(아카이버 프로세스)	14-27
아카이브 로그 파일: 이름 지정 및 대상	14-28
ARCHIVELOG 모드 활성화	14-29
퀴즈	14-30
요약	14-32
연습 14 개요: Recovery 가능성을 위한 구성	14-33

## 15 데이터베이스 백업 수행

목표	15-2
백업 솔루션: 개요	15-3
Oracle Secure Backup	15-4
User-Managed 방식의 백업	15-5
용어	15-6
RMAN(Recovery Manager)	15-8
백업 설정 구성	15-9
백업 일정 잡기: 전략	15-11
백업 일정 잡기: 옵션	15-12
백업 일정 잡기: 설정	15-13
백업 일정 잡기: 일정	15-14
백업 일정 잡기: 검토	15-15
콘트롤 파일을 Trace File로 백업	15-16
백업 관리	15-17
백업 보고서 보기	15-18
Fast Recovery Area 모니터	15-19
RMAN 명령행 사용	15-20
퀴즈	15-21
요약	15-22
연습 15 개요: 데이터베이스 백업 생성	15-23

## 16 데이터베이스 Recovery 수행

목표	16-2
데이터베이스 열기	16-3
데이터베이스를 열린 상태로 유지	16-5
Data Recovery Advisor	16-6
콘트롤 파일 손실	16-8
리두 로그 파일 손실	16-9
NOARCHIVELOG 모드에서의 데이터 파일 손실	16-11

ARCHIVELOG 모드에서의 Noncritical 데이터 파일 손실	16-12
ARCHIVELOG 모드에서의 시스템 Critical 데이터 파일 손실	16-13
Data Failure: 예제	16-14
Data Recovery Advisor	16-15
Data Failure 액세스	16-16
Data Failure	16-17
Data Failure 나열	16-18
복구에 대한 조언	16-19
복구 실행	16-20
Data Recovery Advisor 뷰	16-21
퀴즈	16-22
요약	16-24
연습 16 개요: 데이터베이스 Recovery 수행	16-25

## 17 데이터 이동

목표	17-2
데이터 이동: 일반적 구조	17-3
Oracle Data Pump: 개요	17-4
Oracle Data Pump: 이점	17-5
Data Pump를 위한 디렉토리 객체	17-7
디렉토리 객체 생성	17-8
Data Pump 엑스포트/임포트 클라이언트: 개요	17-9
Data Pump 유ти리티: 인터페이스 및 모드	17-10
Database Control을 사용하여 Data Pump 엑스포트	17-11
Data Pump 엑스포트 예제: 기본 옵션	17-12
Data Pump 엑스포트 예제: 고급 옵션	17-13
Data Pump 엑스포트 예제: 파일	17-14
Data Pump 엑스포트 예제: 일정	17-16
Data Pump 엑스포트 예제: 검토	17-17
Data Pump 임포트 예제: impdp	17-18
Data Pump 임포트: 변형	17-19
Enterprise Manager를 사용하여 Data Pump 작업 모니터	17-20
Data Pump 기존 모드를 사용하여 이전	17-21
Data Pump 기존 모드	17-22
파일 위치 관리	17-24
SQL*Loader: 개요	17-25
SQL*Loader로 데이터 로드	17-27
SQL*Loader 콘트롤 파일	17-28

로드 방식	17-30
External Table	17-31
External Table 이점	17-32
ORACLE_LOADER 로 External Table 정의	17-33
ORACLE_DATAPUMP 로 External Table 채우기	17-34
External Table 사용	17-35
데이터 덕셔너리	17-36
퀴즈	17-37
요약	17-39
연습 17 개요: 데이터 이동	17-40

## 18 지원 활용

목표	18-2
Support Workbench 사용	18-3
Enterprise Manager에서 Critical 오류 Alert 보기	18-4
문제 세부 정보 보기	18-5
Incident 세부 정보 보기: 덤프 파일	18-6
Incident 세부 정보 보기: 검사 프로그램 실행 결과	18-7
서비스 요청 생성	18-8
진단 데이터를 패키지화하여 Oracle 지원 센터에 업로드	18-9
서비스 요청 추적 및 복구 구현	18-10
Incident 및 문제 닫기	18-12
Incident 패키지 구성	18-13
ASM 용 Enterprise Manager Support Workbench	18-14
Oracle 지원 센터 활용	18-15
My Oracle Support 통합	18-16
My Oracle Support 사용	18-17
문제 조사	18-19
서비스 요청 로깅	18-21
패치 관리	18-23
패치 릴리스 적용	18-24
Patch Advisor 사용	18-25
Patch Wizard 사용	18-26

패치 적용	18-27
패치 스테이지	18-28
온라인 패치: 개요	18-29
온라인 패치 설치	18-30
온라인 패치의 이점	18-31
일반 패치 및 온라인 패치	18-32
온라인 패치 고려 사항	18-33
퀴즈	18-35
요약	18-36
연습 18 개요: Alert 및 패치에 EM 도구 사용	18-37

## 부록 A: 연습 및 해답

## 부록 B: 기본 Linux 및 vi 명령

## 부록 C: SQL 문 구문

## 부록 D: Oracle 백그라운드 프로세스

## 부록 E: 머리 글자어 및 용어

### F Oracle Restart

목표	F-2
Oracle Restart	F-3
Oracle Restart 프로세스 시작	F-5
Oracle Restart 제어	F-6
올바른 SRVCTL 유ти리티 선택	F-8
Oracle Restart 구성	F-9
SRVCTL 유ти리티 사용	F-10
SRVCTL 유ти리티에 대한 도움말 보기	F-11
SRVCTL 유ти리티를 사용하여 구성 요소 시작	F-12
SRVCTL 유ти리티를 사용하여 구성 요소 정지	F-13
구성 요소 상태 보기	F-14
구성 요소에 대한 Oracle Restart 구성 표시	F-15
Oracle Restart 구성에 구성 요소를 수동으로 추가	F-16
퀴즈	F-17
요약	F-18
연습 3-1: 개요	F-19

## G 지속적인 교육 및 추가 정보

- 후속 학습 G-2
- 지속적인 교육 자료 G-3
- Oracle University G-4
- 지속적인 교육 G-5
- 데이터베이스 전문 영역 G-6
- Oracle Real Application Clusters G-7
- Oracle Data Guard G-8
- Streams 개요 G-9
- Oracle Technology Network G-11
- 보안 G-12
- Oracle by Example G-13
- Oracle Magazine G-14
- Oracle Applications 커뮤니티 G-15
- 기술 지원: My Oracle Support G-16
- Oracle Database 제품 페이지 G-17
- 감사의 말 G-18

# I

## 소개

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 과정 목표

이 과정을 마치면 다음을 수행할 수 있습니다.

- Oracle Database 11g Release 2 설치, 생성 및 관리
- 응용 프로그램에 대해 데이터베이스 구성
- 기본 모니터 절차 사용
- 백업 및 recovery 전략 구현
- 데이터베이스 및 파일 간 데이터 이동

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 과정 목표

본 과정에서는 Oracle Database 11g Release 2 Enterprise Edition 소프트웨어를 설치하고 새 데이터베이스를 생성한 후 데이터베이스를 관리하는 방법을 배웁니다.

또한 응용 프로그램을 지원하도록 데이터베이스를 구성하고 유저 생성, 저장 영역 구조 정의 및 보안 설정 등의 작업을 수행합니다. 본 과정에서는 가상 응용 프로그램을 사용하지만 실제 응용 프로그램에 필요한 핵심 작업을 모두 수행해 볼 것입니다.

데이터베이스 구성이 완료되었다고 데이터베이스 관리가 끝나는 것은 아닙니다.

본 과정에서는 백업 및 recovery 전략을 설계하여 데이터베이스를 보호하는 방법 및 데이터베이스가 올바로 작동하도록 모니터하는 방법에 대해서도 설명합니다.

# 제안 일정

일	단원	일	단원		
1	1. 오라클 데이터베이스 구조 탐색 2. 데이터베이스 환경 준비 3. 오라클 데이터베이스 생성 4. 데이터베이스 Instance 관리	3	9. 데이터 동시성 관리 10. 언두 데이터 관리 11. 오라클 데이터베이스 감사(audit) 구현		
2	5. ASM Instance 관리 6. Oracle 네트워크 환경 구성 7. 데이터베이스 저장 영역 구조 관리 8. 유저 보안 관리	4	12. 데이터베이스 유지 관리 13. 성능 관리 14. 백업 및 Recovery 개념	5	15. 데이터베이스 백업 수행 16. 데이터베이스 Recovery 수행 17. 데이터 이동 18. 지원 활용

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 오라클 제품 및 서비스

- **오라클 데이터베이스**
- **Oracle WebLogic Application Server**
- **Oracle Applications**
- **Oracle Collaboration Suite**
- **Oracle Developer Suite**
- **오라클 서비스**



**ORACLE®**

**ORACLE®**

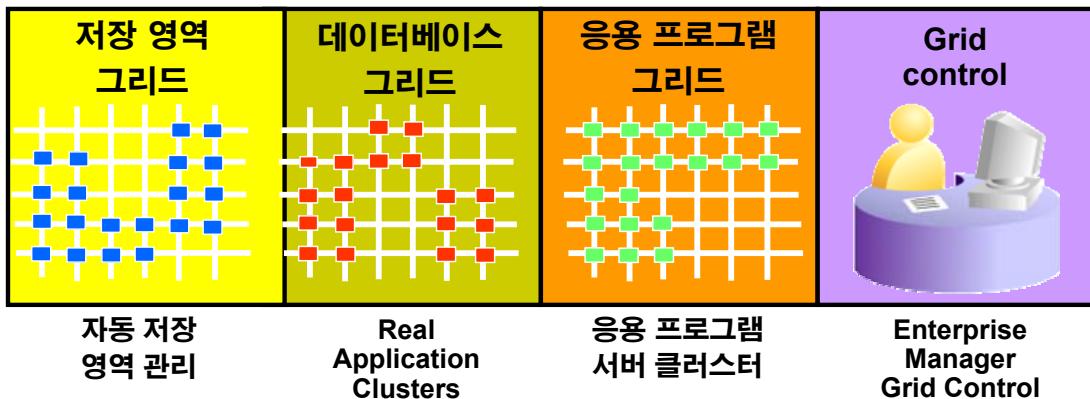
Copyright © 2010, Oracle. All rights reserved.

## 오라클 제품 및 서비스

- **오라클 데이터베이스:** 오라클 데이터베이스는 정보 및 응용 프로그램을 관리하는 가장 유연하면서도 경제적인 방법인 엔터프라이즈 그리드 컴퓨팅으로 설계된 첫번째 데이터베이스입니다.
- **Oracle WebLogic Application Server:** 웹 기반 응용 프로그램을 개발 및 배치하는 데 필요한 모든 기술을 통합한 오라클의 Java 2 Platform, Enterprise Edition 인증 서버입니다. 이 Application Server는 PL/SQL, Oracle Forms 및 Java EE 기반 응용 프로그램 등 트랜잭션 응용 프로그램, 웹 서비스 및 e-business 포털을 배치합니다.
- **Oracle Applications:** Oracle E-Business Suite는 조직 전체에서 프로세스를 관리 및 자동화하기 위한 완전한 업무용 응용 프로그램입니다.
- **Oracle Collaboration Suite:** Oracle Collaboration Suite는 음성, 전자 메일, 팩스, 무선, 달력 정보 및 파일과 같은 조직의 모든 통신 데이터에 대한 단일 통합 시스템입니다.
- **Oracle Developer Suite:** Oracle Developer Suite는 응용 프로그램 개발과 Business Intelligence 도구를 결합하는 완전한 통합 환경입니다.
- **오라클 서비스:** Oracle Consulting 및 Oracle University와 같은 서비스를 통해 오라클 프로젝트에 필요한 전문 지식을 제공합니다. 다양한 교육 자료에 대한 링크를 보려면 부록 "후속 단계: 지속적인 교육"을 참조하십시오.

# Oracle Database 11g: "g"는 그리드(Grid)의 약어

- OGF(Open Grid Forum)
- 오라클의 그리드 Infrastructure:
  - 낮은 비용
  - 높은 서비스 품질
  - 편리한 관리



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Oracle Database 11g: "g"는 그리드(Grid)의 약어

OGF(Open Grid Forum)는 그리드 컴퓨팅의 표준을 개발하는 표준 단체입니다. 이 단체는 그리드 컴퓨팅의 다양한 측면을 다루는 여러 위원회와 워크그룹으로 구성되어 있습니다. 위원회와 워크그룹은 교육 기관, 리서치 커뮤니티 및 일반 회사(점차 증가하는 추세)의 참가자들로 구성됩니다. <http://www.ogf.org>에서 OGF 웹 사이트를 볼 수 있습니다.

오라클에서는 모든 유형의 작업 로드에 대한 서버 간 균형을 유지하고 이 모든 서버를 하나의 완벽한 시스템으로 관리할 수 있는 그리드 컴퓨팅 Infrastructure 소프트웨어를 만들었습니다. 그리드 컴퓨팅에서는 모든 구성 요소가 클러스터화되기 때문에 메인프레임 컴퓨팅과 동일한 수준의 매우 높은 신뢰성을 달성할 수 있습니다. 그러나 메인프레임이나 대형 UNIX SMP(Symmetric Multiprocessing) 서버와 달리 그리드는 Intel 프로세서 및 Linux 운영 체제와 같은 개방형 시스템 기술을 사용하므로 매우 적은 비용으로 구축할 수 있습니다.

오라클의 그리드 컴퓨팅 기술에는 다음과 같은 사항이 포함됩니다.

- 자동 저장 영역 관리(ASM)
- RAC(Real Application Clusters)
- Application Server 클러스터
- Enterprise Manager Grid Control

## Oracle Database 11g: "g"는 그리드(Grid)의 약어(계속)

**자동 저장 영역 관리(ASM)**은 모든 디스크에 데이터베이스 데이터를 분배하고, 저장 영역 그리드를 생성 및 유지 관리하며, 최소 관리 비용으로 가장 높은 수준의 입출력(I/O) 처리 능력을 제공합니다. 디스크가 추가되거나 삭제되면 ASM은 데이터를 자동으로 재분배합니다. 따라서 파일 시스템을 관리하기 위한 논리적 볼륨 관리자가 필요하지 않습니다. 선택 사항인 **mirroring**을 사용하면 데이터 가용성이 증가되며 온라인으로 디스크를 추가하거나 삭제할 수 있습니다. 자세한 사항은 "데이터베이스 저장 영역 구조 관리" 단원을 참조하십시오.

오라클의 **Real Application Clusters**는 서버 클러스터에서 모든 응용 프로그램 작업 로드를 실행하고 조정하며 다음과 같은 기능을 제공합니다.

- **통합 클러스터 웨어:** 여기에는 클러스터 연결, 메시징 및 lock, 클러스터 제어 및 recovery 기능이 포함됩니다. 이 기능은 Oracle Database 10g 이상에서 지원되는 모든 플랫폼에서 제공됩니다.
- **자동 작업 로드 관리:** 정상적인 작업 중 또는 failure에 대한 응답으로 각 서비스에 프로세싱 리소스를 자동으로 할당하는 규칙을 정의할 수 있습니다. 이러한 규칙은 변화하는 업무 요구 사항에 부합하도록 동적으로 수정될 수 있습니다. 데이터베이스 그리드 내에서 동적으로 리소스를 할당하는 기능은 Oracle RAC에서만 볼 수 있는 기능입니다.
- **Automatic event notification to the mid-tier:** 클러스터 구성이 변경되면 mid-tier가 즉시 instance failover 또는 사용 가능한 새 instance를 적용할 수 있습니다. 따라서 instance failover 상황에서 일반 유저가 네트워크 타임아웃으로 인한 지연 없이 계속 작업을 수행할 수 있습니다. 새 instance를 사용할 수 있게 되면 mid-tier가 즉시 해당 instance로 로드 밸런싱 연결을 시작할 수 있습니다. Oracle Database 10g 또는 이후 JDBC(Java Database Connectivity) 드라이버에는 이러한 이벤트를 자동으로 처리할 수 있는 "Fast Connection Failover" 기능이 있습니다.

**Oracle WebLogic Application Grid**는 Oracle WebLogic Server, IBM WebSphere Application Server 및 JBoss Application Server를 비롯한 모든 Application Server에서 작동하며 Application Server가 없는 순수한 그리드 환경에서도 작동합니다. Oracle WebLogic Application Grid는 매우 뛰어나고 예측 가능한 응용 프로그램 확장성 및 성능을 제공합니다. 요구 시 용량을 확장할 수 있는 Oracle WebLogic Application Grid는 몇 개의 서버에서 수 천 개의 서버까지 미들웨어 infrastructure를 선형적으로 확장할 수 있습니다. in-memory 데이터 그리드 솔루션을 통해 자주 사용되는 데이터에 대한 신속한 액세스를 제공합니다. 이러한 그리드 성능을 활용하여 계산을 병렬로 수행함으로써 응용 프로그램 성능을 더욱 향상시켜 줍니다.

**Enterprise Manager Grid Control**은 전체 소프트웨어 스택 관리, 유저 프로비저닝(provisioning), 데이터베이스 복제(Database Cloning) 및 패치 관리를 포함하여 그리드 전체에 적용되는 작업을 관리합니다. 여기에서는 일반 유저의 관점에서 모든 응용 프로그램의 성능을 모니터할 수 있습니다. Grid Control에서는 그리드 Infrastructure의 성능 및 가용성을 분리된 저장 영역 단위, 데이터베이스 및 Application Server의 관점에서 보는 것이 아니라 통합된 하나의 모습으로 봅니다. 하드웨어 노드, 데이터베이스 및 Application Server를 단일 논리적 엔티티로 그룹화하고 대상 그룹을 하나의 단위로 관리할 수 있습니다.

**주:** 본 과정에서는 Enterprise Manager Database Console을 사용하여 한 번에 하나의 데이터베이스를 관리합니다.

## 단일 Instance용 그리드 Infrastructure

**Oracle Database 11g Release 2에서는 단일 Instance용 그리드 Infrastructure를 선보입니다.**

- 오라클 데이터베이스 소프트웨어와는 별개로 클러스터웨어 미디어에서 설치됨
- Oracle 자동 저장 영역 관리(ASM) 포함
- Oracle Restart(클러스터화되지 않은 데이터베이스를 위한 고가용성 솔루션) 포함
  - 다음 구성 요소를 모니터 및 재시작할 수 있음:
    - 데이터베이스 Instance
    - Oracle Net 리스너
    - 데이터베이스 서비스
    - 자동 저장 영역 관리(ASM) Instance
    - ASM 디스크 그룹
    - Data Guard용 Oracle Notification Services(ONS/eONS)

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 단일 Instance용 그리드 Infrastructure

Oracle Database 11g Release 2에서는 단일 Instance용 그리드 Infrastructure를 선보입니다. 이 Infrastructure는 오라클 데이터베이스 소프트웨어와는 별개로 클러스터웨어 미디어에서 설치되며, Oracle 자동 저장 영역 관리(ASM) 및 새로운 Oracle Restart라는 새로운 기능을 포함합니다.

Oracle Restart는 오라클 데이터베이스의 가용성을 향상시킬 수 있도록 설계되었습니다. 이 기능은 단일 Instance(클러스터화되지 않은) 환경에 대해서만 고가용성 솔루션을 구현합니다. Oracle RAC(Real Application Cluster) 환경의 경우 구성 요소를 자동으로 재시작하는 기능은 Oracle Clusterware에서 제공됩니다. Oracle Restart는 다음 구성 요소의 상태를 모니터하고 이를 자동으로 재시작할 수 있습니다.

- 데이터베이스 Instance
- Oracle Net 리스너
- 데이터베이스 서비스
- ASM Instance
- ASM 디스크 그룹
- Data Guard용 Oracle Notification Services(ONS/eONS)

Oracle Restart는 구성 요소가 구성 요소 종속성에 따라 적합한 순서로 시작되도록 보장합니다. 특정 구성 요소를 종료해야 할 경우 종속된 구성 요소가 먼저 정상적으로 종료되도록 보장합니다. Oracle Restart는 오라클 데이터베이스 홈과 별개로 설치하는 Oracle 그리드 Infrastructure 홈에서 실행됩니다.



# 1 오라클 데이터베이스 구조 탐색

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 오라클 데이터베이스의 주요 구조적 구성 요소 나열
- 메모리 구조 설명
- 백그라운드 프로세스 설명
- 논리적/물리적 저장 영역 구조 상호 연관
- ASM 저장 영역 구성 요소 설명



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

이 단원에서는 오라클 데이터베이스 구조에 대해 자세히 설명합니다. 이러한 설명을 통해 물리적 구조와 논리적 구조 및 다양한 구성 요소에 대해 알아볼 수 있습니다.

# 오라클 데이터베이스

오라클 RDBMS(관계형 데이터베이스 관리 시스템)를 사용하면 정보를 개방적이고 포괄적이며 통합적으로 관리할 수 있습니다.



ORACLE®

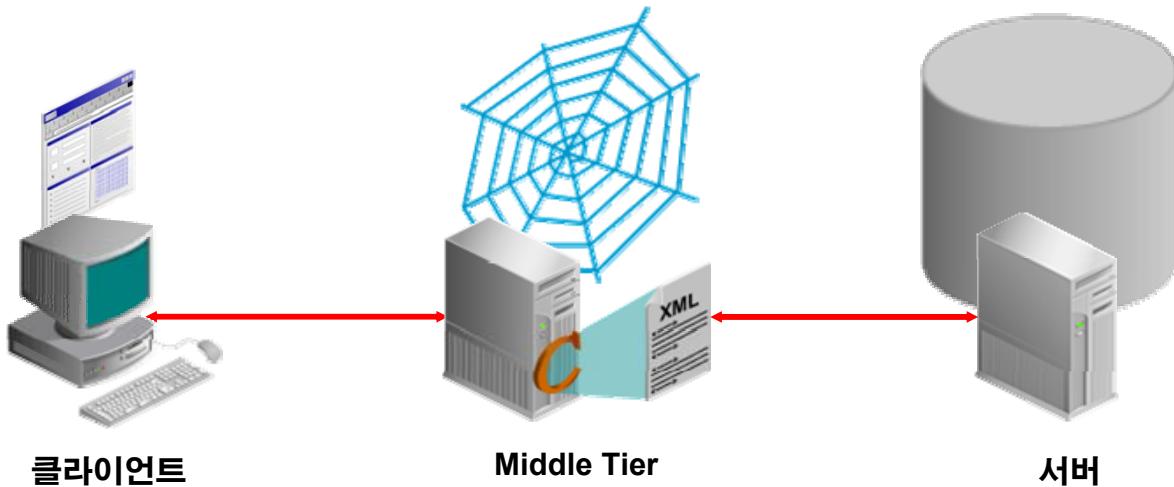
Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스

데이터베이스는 하나의 단위로 취급되는 데이터 모음입니다. 데이터베이스의 목적은 관련 정보를 저장 및 검색하기 위한 것입니다.

오라클 RDBMS(관계형 데이터베이스 관리 시스템)는 다중 유저 환경에서 많은 양의 데이터를 안정적으로 관리하여 다수의 유저가 동일한 데이터에 동시에 액세스할 수 있도록 합니다. 이러한 기능은 뛰어난 성능과 함께 제공됩니다. 또한 오라클 데이터베이스는 권한 없는 액세스를 차단하고 Failure Recovery에 대한 효율적인 솔루션을 제공합니다.

## 서버에 연결



여러 계층 구조를 보여주는 그림

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 서버에 연결

데이터베이스 유저는 다음 세 가지 방법 중 하나로 Oracle 서버에 연결할 수 있습니다.

- 유저가 Oracle Instance를 실행하는 운영 체제에 로그온하고 이 시스템에 있는 데이터베이스에 액세스하는 응용 프로그램 또는 도구를 시작합니다. 통신 경로는 호스트 운영 체제에서 사용 가능한 프로세스 간 통신 메커니즘을 사용하여 설정됩니다.
- 로컬 컴퓨터에서 응용 프로그램이나 도구를 시작한 다음 네트워크를 통해 오라클 데이터베이스를 실행하는 컴퓨터에 연결합니다. 클라이언트/서버라고 하는 이 구성에서 네트워크 소프트웨어는 유저와 백 엔드 서버 사이의 통신에 사용됩니다. 클라이언트/서버 구조 데이터베이스 시스템은 네트워크를 통해 연결된 Front-End(클라이언트) 및 백 엔드(서버)의 두 부분으로 구성됩니다. 네트워크 소프트웨어는 유저와 Oracle 서버 간의 통신에 사용됩니다.
  - 클라이언트는 데이터베이스 서버에서 수행할 작업에 대한 요청을 시작하는 데이터베이스 응용 프로그램입니다. 클라이언트는 서버에서 관리하는 데이터를 요청, 처리 및 제공합니다. 클라이언트 워크스테이션은 해당 작업용으로 최적화할 수 있습니다. 예를 들어, 클라이언트는 대규모 디스크 용량을 필요로 하지 않을 수도 있고 그래픽 기능을 활용할 수도 있습니다. 클라이언트는 데이터베이스 서버보다 다른 컴퓨터에서 실행되는 경우가 많습니다. 여러 개의 클라이언트를 하나의 서버에 대해 동시에 실행할 수 있습니다.

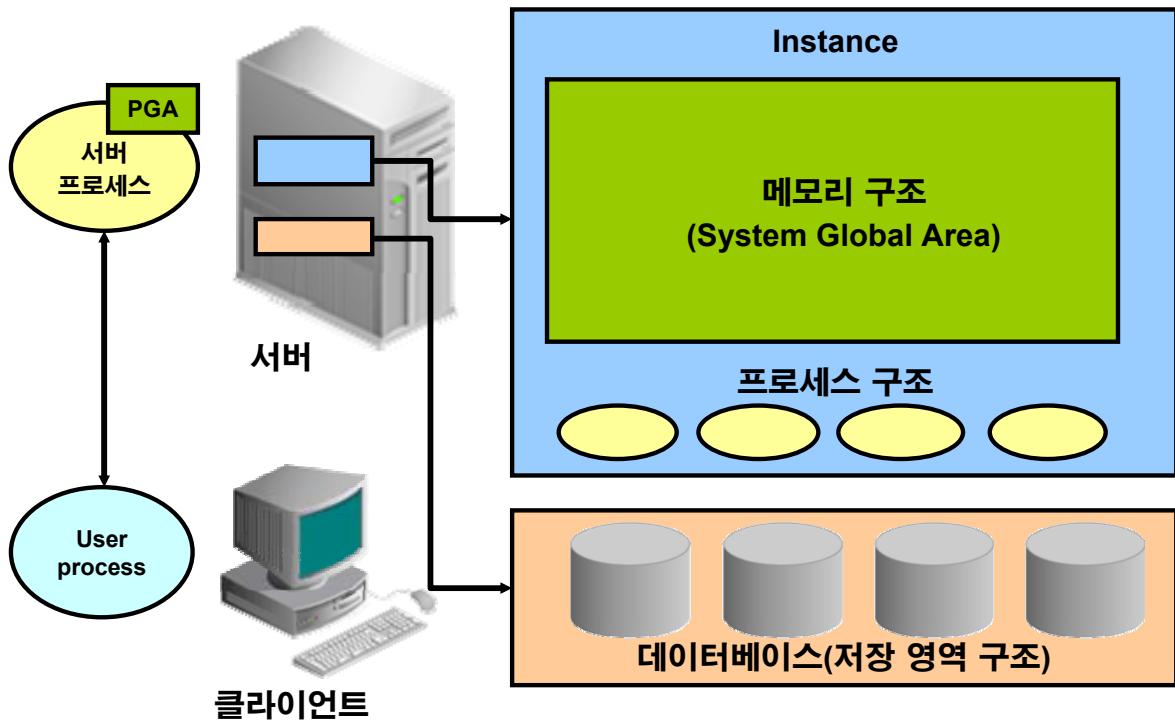
## 서버에 연결(계속)

- 서버는 오라클 데이터베이스 소프트웨어를 실행하며 동시 공유 데이터 액세스에 필요한 기능을 처리합니다. 서버는 클라이언트 응용 프로그램에서 생성되는 요청을 수신 및 처리합니다. 서버를 관리하는 컴퓨터는 해당 작업용으로 최적화할 수 있습니다. 예를 들어, 서버 컴퓨터는 고용량 디스크와 고속 프로세서를 포함할 수 있습니다.
- 유저는 로컬 컴퓨터(클라이언트)에서 웹 브라우저 등의 도구를 통해 Application Server에 액세스합니다. 그런 다음 Application Server는 클라이언트를 대신하여 백 엔드 데이터베이스 서버와 상호 작용합니다.

일반적인 여러 계층 구조에는 다음과 같은 구성 요소가 포함됩니다.

- 작업을 시작하는 클라이언트 또는 시작 프로그램 프로세스
  - 작업의 일부를 수행하는 하나 이상의 Application Server. Application Server는 대부분의 응용 프로그램 논리를 포함하고, 클라이언트에 대한 데이터 액세스를 제공하며, 일부 Query 처리를 수행하여 데이터베이스 서버의 일부 로드를 제거합니다. Application Server는 클라이언트와 여러 데이터베이스 서버 간의 인터페이스 역할을 할 수 있으며, 추가 보안 레벨을 제공할 수 있습니다.
  - 작업에 사용되는 대부분의 데이터를 저장하는 엔드 서버 또는 데이터베이스 서버
- 이 구조에서는 Application Server를 사용하여 다음을 수행할 수 있습니다.
- 웹 브라우저 등의 클라이언트 인증서 검증
  - 오라클 데이터베이스 서버에 연결
  - 클라이언트를 대신하여 요청된 작업 수행

## 오라클 데이터베이스 서버 구조: 개요



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 오라클 데이터베이스 서버 구조

오라클 데이터베이스 서버 아키텍처에서 중요한 세 가지 구조는 메모리 구조, 프로세스 구조 및 저장 영역 구조입니다. 기본 오라클 데이터베이스 시스템은 오라클 데이터베이스와 데이터베이스 Instance로 구성됩니다.

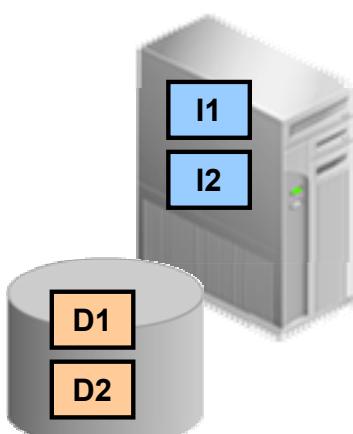
데이터베이스에는 물리적 구조와 논리적 구조로 구성됩니다. 물리적 구조와 논리적 구조는 서로 별개의 구조이기 때문에 논리적 저장 영역 구조에 대한 액세스에 영향을 주지 않고 데이터의 물리적 저장 영역을 관리할 수 있습니다.

Instance는 메모리 구조 및 해당 Instance와 연관된 백그라운드 프로세스로 구성됩니다. Instance가 시작될 때마다 SGA(System Global Area)라는 공유 메모리 영역이 할당되고 백그라운드 프로세스가 시작됩니다. 프로세스는 컴퓨터의 메모리에서 수행되는 작업입니다. 프로세스는 "제어 스레드" 또는 운영 체제에서 일련의 단계로 실행될 수 있는 메커니즘으로 정의됩니다. Oracle 소프트웨어는 데이터베이스 Instance를 시작한 후 해당 Instance를 특정 데이터베이스와 연관시킵니다. 이를 데이터베이스를 마운트한다고 부릅니다. 그런 다음 데이터베이스를 열 준비가 되고 승인된 유저가 액세스할 수 있는 상태가 됩니다.

주: Oracle 자동 저장 영역 관리(ASM)는 메모리 및 프로세스 구성 요소에 대해 Instance의 개념을 사용하지만 특정 데이터베이스와 연관되지는 않습니다.

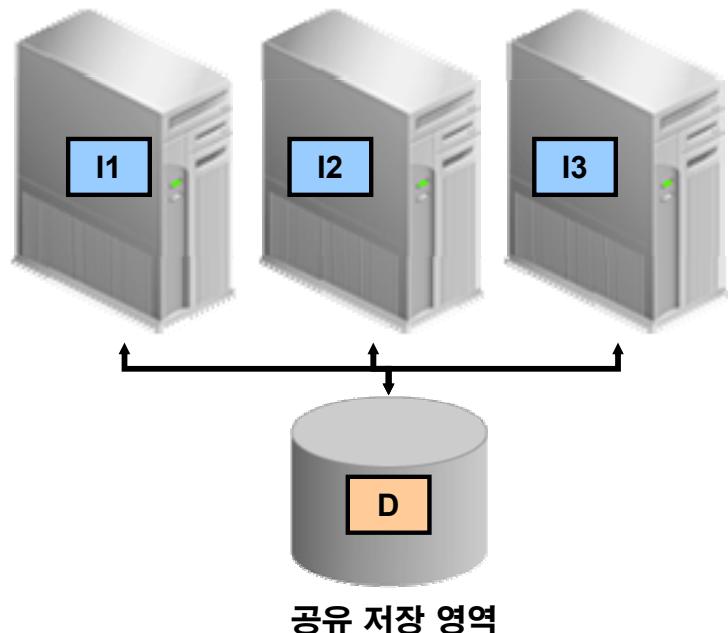
## Instance: 데이터베이스 구성

클러스터화되지 않은 시스템



로컬 저장 영역

클러스터화된 시스템



ORACLE®

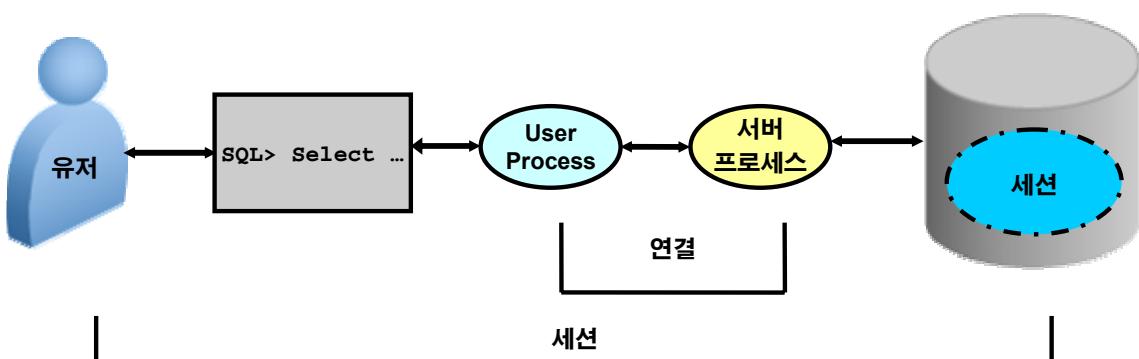
Copyright © 2010, Oracle. All rights reserved.

### Instance: 데이터베이스 구성

각 데이터베이스 Instance는 단 한 개의 데이터베이스와만 연관됩니다. 동일 서버에 데이터베이스가 여러 개 있는 경우 각 데이터베이스마다 별개의 고유한 데이터베이스 Instance가 존재합니다. 데이터베이스 Instance는 공유될 수 없습니다. RAC(Real Applications Cluster) 데이터베이스에는 일반적으로 여러 개별 서버에 동일 공유 데이터베이스에 대한 여러 Instance가 포함됩니다. 이 모델에서는 동일 데이터베이스가 각 RAC Instance와 연관되어 하나의 Instance에 대해 최대 하나의 데이터베이스만 연관시켜야 하는 필요 조건을 충족시킵니다.

# 데이터베이스 Instance에 연결

- **연결:** User Process와 Instance 간의 통신
- **세션:** 유저가 User Process를 통해 수행하는 Instance에 대한 특정 연결



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 Instance에 연결

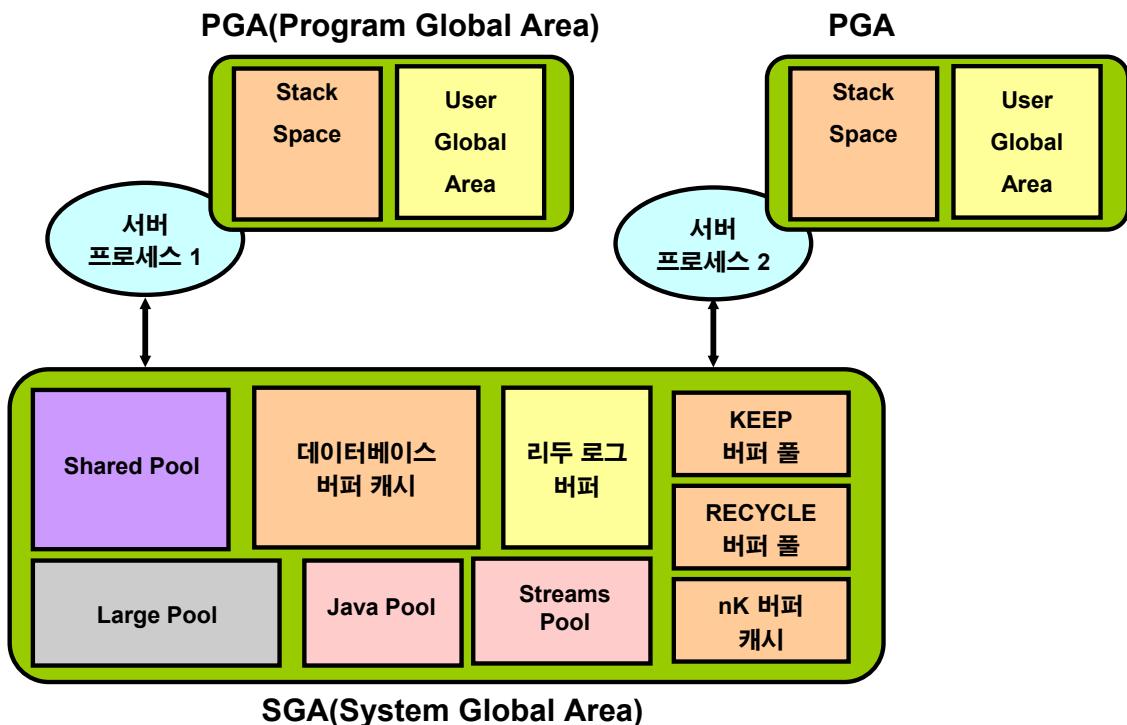
연결 및 세션은 User Process와 밀접하게 연관되어 있지만 그 의미는 전혀 다릅니다.

연결은 User Process와 오라클 데이터베이스 Instance 사이의 통신 경로입니다. 통신 경로는 한 컴퓨터에서 User Process 및 오라클 데이터베이스를 모두 실행하는 경우 사용 가능한 프로세스 간 통신 메커니즘을 통해 설정되며, 서로 다른 컴퓨터에서 데이터베이스 응용 프로그램 및 오라클 데이터베이스를 실행하며 네트워크를 통해 통신하는 경우 네트워크 소프트웨어를 통해 설정됩니다.

세션은 데이터베이스 Instance에 대한 현재 유저의 로그인 상태를 나타냅니다. 예를 들어, 유저가 SQL\*Plus를 시작하는 경우 적합한 Username과 암호를 제공해야 합니다. 그러면 해당 유저에 대한 세션이 설정됩니다. 세션은 유저가 연결하는 시점에서 데이터베이스 응용 프로그램 연결을 끊거나 데이터베이스 응용 프로그램을 종료하는 시점까지 지속됩니다.

동일한 Username을 사용하는 단일 오라클 데이터베이스 유저에 대해 여러 세션이 동시에 생성되고 존재할 수 있습니다. 예를 들어, Username/암호가 HR/HR인 유저가 동일한 오라클 데이터베이스 Instance에 여러 번 연결할 수 있습니다.

## 오라클 데이터베이스 메모리 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 오라클 데이터베이스 메모리 구조

오라클 데이터베이스는 다양한 용도로 메모리 구조를 생성 및 사용합니다. 예를 들어, 메모리에는 실행 중인 프로그램 코드, 유저 간에 공유되는 데이터 및 각 연결된 유저의 전용(Private) 데이터 영역이 저장됩니다.

Instance에는 다음과 같은 두 가지 기본 메모리 구조가 연관되어 있습니다.

- **SGA(System Global Area):** 하나의 오라클 데이터베이스 Instance의 데이터 및 제어 정보를 포함하는 공유 메모리 구조(SGA 구성 요소)의 그룹입니다. SGA는 모든 서버 및 백그라운드 프로세스에서 공유됩니다. SGA에 저장된 데이터의 예로는 캐시된 데이터 블록 및 공유 SQL 영역이 있습니다.
- **PGA(Program Global Areas):** 서버 또는 백그라운드 프로세스의 데이터 및 제어 정보를 포함하는 메모리 영역입니다. PGA는 서버 또는 백그라운드 프로세스를 시작할 때 오라클 데이터베이스에서 생성되는 비공유 메모리입니다. PGA에는 서버 프로세스만 액세스할 수 있습니다. 서버 프로세스와 백그라운드 프로세스는 각각 자체적인 PGA를 갖습니다.

## 오라클 데이터베이스 메모리 구조(계속)

SGA는 Instance에 대한 데이터 및 제어 정보를 포함하는 메모리 영역입니다. SGA는 다음 데이터 구조를 포함합니다.

- **Shared Pool:** 유저 간에 공유할 수 있는 다양한 구성요소를 캐시에 저장합니다.
- **데이터베이스 버퍼 캐시:** 데이터베이스에서 검색된 데이터 블록을 캐시에 저장합니다.
- **KEEP 버퍼 풀:** 메모리에 데이터 블록을 오랫동안 보존하도록 튜닝된 특별한 유형의 데이터베이스 버퍼 캐시입니다.
- **RECYCLE 버퍼 풀:** 메모리에서 블록을 신속하게 재사용 또는 제거하도록 튜닝된 특별한 유형의 데이터베이스 버퍼 캐시입니다.
- **nK 버퍼 캐시:** 기본 데이터베이스 블록 크기와 다른 블록 크기를 저장하도록 설계된 몇 가지 특별한 유형의 데이터베이스 버퍼 캐시 중 하나입니다.
- **리두 로그 버퍼:** Instance Recovery에 사용되는 리두 정보가 디스크에 저장된 물리적 리두 로그 파일에 기록될 때까지 해당 정보를 캐시에 저장합니다.
- **Large Pool:** 오라클 백업 및 Recovery 작업과 같은 특정 대용량 프로세스 및 I/O 서버 프로세스에 대한 대용량 메모리 할당을 제공하는 선택적 영역입니다.
- **Java Pool:** JVM(Java Virtual Machine) 내의 모든 세션별 Java 코드 및 데이터에 사용됩니다.
- **Streams Pool:** Oracle Streams에서 캡처 및 적용에 필요한 정보를 저장하는 데 사용됩니다.

Enterprise Manager 또는 SQL\*Plus를 사용하여 Instance를 시작하면 SGA에 대해 할당된 메모리 양이 표시됩니다.

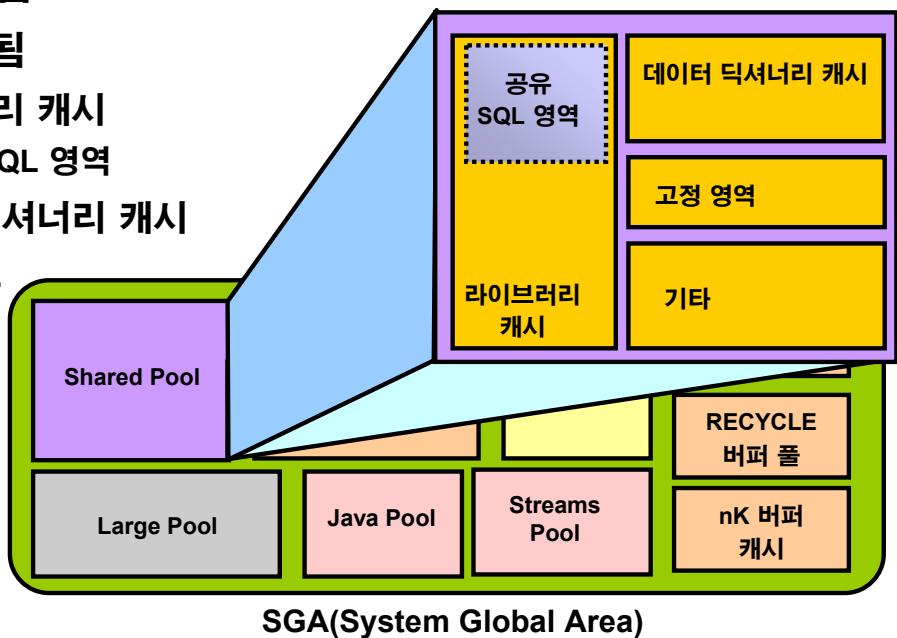
PGA(Program Global Area)는 각 서버 프로세스에 대한 데이터 및 제어 정보를 포함하는 메모리 영역입니다. Oracle 서버 프로세스는 클라이언트의 요청을 처리합니다. 각 서버 프로세스에는 서버 프로세스가 시작될 때 할당되는 자체 전용(Private) PGA가 있습니다. PGA에는 해당 서버 프로세스만 액세스할 수 있으며, 서버 프로세스를 대신하는 Oracle 코드에 의해서만 PGA를 읽고 쓸 수 있습니다. PGA는 Stack Space 및 UGA(User Global Area)의 두 주요 영역으로 구분됩니다.

동적 SGA Infrastructure를 사용하면 Instance를 종료하지 않고 데이터베이스 버퍼 캐시, Shared Pool, Large Pool, Java Pool 및 Streams Pool의 크기를 변경할 수 있습니다.

오라클 데이터베이스는 초기화 파라미터를 사용하여 메모리 구조를 생성 및 관리합니다. 메모리를 관리하는 가장 간단한 방법은 데이터베이스가 메모리를 자동으로 관리 및 튜닝하도록 하는 것입니다. 그러면 대부분의 플랫폼에서 목표 메모리 크기 초기화 파라미터(MEMORY\_TARGET)와 최대 메모리 크기 초기화 파라미터(MEMORY\_MAX\_TARGET)만 설정하면 됩니다.

# Shared Pool

- SGA의 일부임
- 다음이 포함됨
  - 라이브러리 캐시
    - 공유 SQL 영역
  - 데이터 딕셔너리 캐시
  - 제어 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Shared Pool

SGA의 Shared Pool 부분에는 라이브러리 캐시, 데이터 딕셔너리 캐시, SQL Query 결과 캐시, PL/SQL 함수 결과 캐시, 병렬 실행 메시지의 버퍼 및 제어 구조가 포함됩니다.

데이터 딕셔너리는 데이터베이스, 해당 구조 및 유저에 대한 참조 정보를 포함하는 데이터베이스 테이블 및 뷰 모음입니다. 오라클 데이터베이스는 SQL 문 구문 분석 중에 데이터 딕셔너리에 자주 액세스합니다. 오라클 데이터베이스가 지속적으로 작업을 수행하려면 이와 같은 액세스가 반드시 필요합니다.

오라클 데이터베이스가 데이터 딕셔너리에 자주 액세스하므로 메모리에 있는 두 곳의 특수 위치가 딕셔너리 데이터를 보관하도록 지정됩니다. 이 중 한 영역은 데이터 딕셔너리 캐시이며, 전체 데이터 블록을 보관하는 버퍼 대신 데이터를 행으로 보관하므로 행 캐시라고도 합니다. 딕셔너리 데이터를 보관하는 메모리의 다른 한 영역은 라이브러리 캐시입니다. 모든 오라클 데이터베이스 User Process는 데이터 딕셔너리 정보 액세스를 위해 이 두 캐시를 공유합니다.

오라클 데이터베이스는 데이터베이스에서 실행하는 각 SQL 문을 공유 SQL 영역(및 PGA에 보관되는 전용 SQL 영역)으로 나타냅니다. 그리고 두 유저가 같은 SQL 문을 실행하는 경우를 인식하고 해당 유저에 대해 공유 SQL 영역을 재사용합니다.

## Shared Pool(계속)

공유 SQL 영역에는 지정된 SQL 문에 대한 구문 분석 트리 및 실행 계획이 포함됩니다. 오라클 데이터베이스는 여러 번 실행되는 SQL 문에 대해 하나의 공유 SQL 영역을 사용하여 메모리를 절약합니다. 이러한 경우는 여러 유저가 동일한 응용 프로그램을 실행하는 경우에 해당합니다.

새 SQL 문이 구문 분석되면 오라클 데이터베이스는 공유 SQL 영역에 새 SQL 문을 저장하기 위해 Shared Pool의 메모리를 할당합니다. 이 메모리의 크기는 명령문의 복합성에 따라 달라집니다.

오라클 데이터베이스는 개별 SQL 문을 처리하는 것과 비슷하게 PL/SQL 프로그램 단위(프로시저, 함수, 패키지, 익명 블록 및 데이터베이스 트리거)를 처리합니다. 또한 프로그램 단위의 구문 분석 및 컴파일된 형식을 보관하기 위한 공유 영역을 할당합니다. 오라클 데이터베이스는 로컬 변수, Global 변수, 패키지 변수(패키지 인스턴스화라고도 함)를 포함한 프로그램 단위를 실행하는 세션 및 SQL 실행을 위한 버퍼와 관련된 값을 보관하기 전용(Private) 영역을 할당합니다. 여러 유저가 동일한 프로그램 단위를 실행하는 경우 모든 유저에게 단일 공유 영역이 사용되며, 모든 유저는 자신의 고유 세션에 특정된 값이 보관된 고유한 전용(Private) SQL 영역의 개별 복사본을 유지 관리합니다.

PL/SQL 프로그램 단위에 포함된 개별 SQL 문은 다른 SQL 문과 같은 방식으로 처리됩니다. PL/SQL 프로그램 단위의 원래 값과는 달리 이러한 SQL 문은 공유 영역을 사용하여 구문 분석된 표현을 보관하고 명령문을 실행하는 각 세션에 대해 전용(Private) 영역을 사용합니다.

SQL Query 결과 캐시 및 PL/SQL 함수 결과 캐시는 Oracle Database 11g에서 새롭게 제공되는 기능으로, 동일한 Infrastructure를 공유하고, 동일한 Dynamic Performance 뷰(V\$)에 표시되며, 동일하게 제공되는 패키지를 통해 관리됩니다.

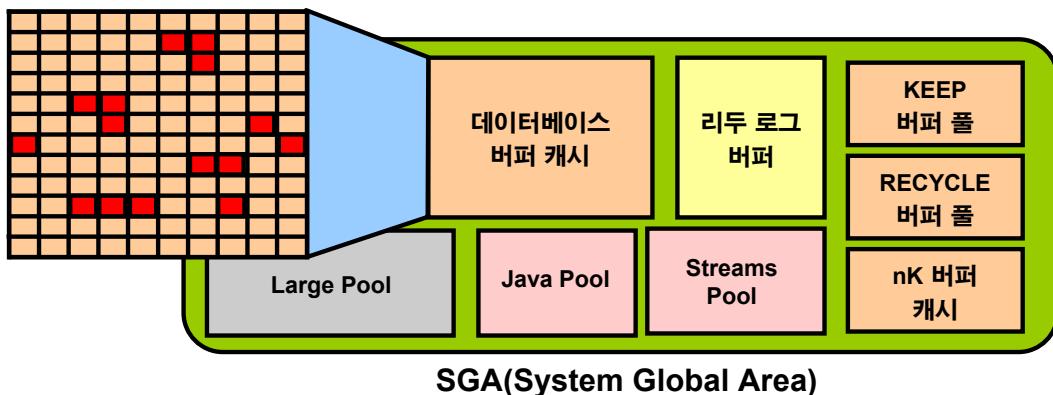
Query 결과 및 Query 부분은 SQL Query 결과 캐시의 메모리에서 캐시에 저장될 수 있습니다. 그러면 데이터베이스가 캐시된 결과를 사용하여 이러한 Query 및 Query 부분의 향후 실행을 수행할 수 있습니다. SQL query 결과 캐시에서 결과를 검색하는 것이 Query를 재실행하는 것보다 빠르기 때문에, 자주 실행하는 Query의 경우 해당 결과가 캐시에 저장되면 성능이 크게 향상됩니다.

PL/SQL 함수는 하나 또는 여러 개의 Parameterized Query를 입력으로 받아들여 함수에서 실행한 계산 결과를 반환하는 데 사용되기도 합니다. 이러한 Query는 함수 호출 빈도에 비해 변경 빈도가 크게 낮은 데이터에 액세스하는 경우도 있습니다. PL/SQL 함수의 소스 텍스트에 구문을 포함하여 해당 결과가 PL/SQL 함수 결과 캐시에 저장되고, 정확성을 유지하기 위해 테이블 리스트의 테이블에 DML이 적용되는 경우 캐시가 지워지도록 요청할 수 있습니다.

Shared Pool의 고정된 영역은 SGA에 대한 시작 오버헤드를 나타냅니다. 크기는 일반적으로 크기가 지정된 Shared Pool 또는 SGA에 비해 매우 작습니다.

## 데이터베이스 버퍼 캐시

- SGA의 일부임
- 데이터 파일에서 읽은 데이터 블록 복사본을 보관함
- 모든 동시에 유저가 공유함



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 버퍼 캐시

데이터베이스 버퍼 캐시는 데이터 파일에서 읽거나 읽기 일관성 모델을 만족시키기 위해 동적으로 생성된 블록 이미지를 보관하는 SGA의 일부입니다. Instance에 동시에 연결하는 모든 유저는 데이터베이스 버퍼 캐시에 대한 액세스를 공유합니다.

오라클 데이터베이스 User Process는 특정 데이터 부분을 처음 사용해야 하는 경우 데이터베이스 버퍼 캐시의 데이터를 검색합니다. 프로세스에서 캐시에 이미 있는 데이터를 발견하는 경우(캐시 적중) 메모리에서 데이터를 직접 읽을 수 있습니다. 프로세스가 캐시의 데이터를 발견하지 못하면(캐시 실패) 데이터에 액세스하기 전에 데이터 블록을 디스크의 데이터 파일에서 캐시의 버퍼로 복사해야 합니다. 캐시 적중을 통한 데이터 액세스가 캐시 실패를 통한 데이터 액세스보다 빠릅니다.

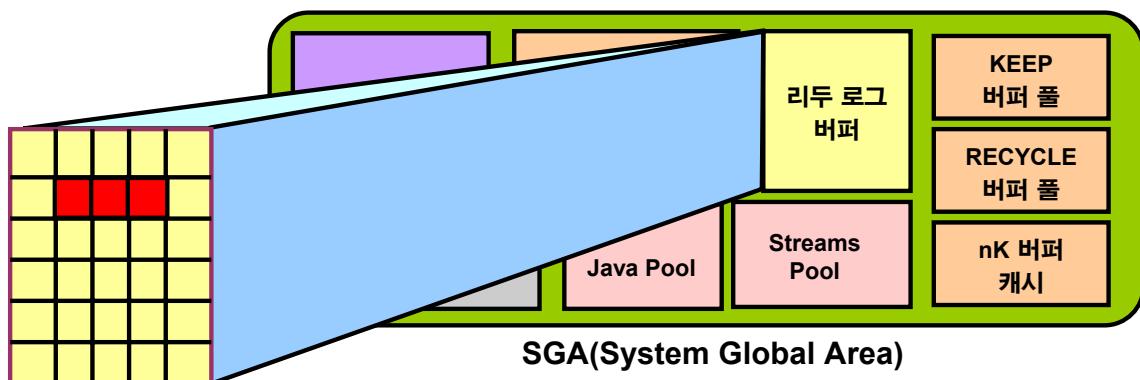
캐시의 버퍼는 LRU(Least Recently Used) List와 접근 횟수의 조합을 사용하는 복합 알고리즘에 의해 관리됩니다. LRU는 디스크 액세스를 최소화하기 위해 최근에 사용된 블록이 메모리에 유지되도록 보장합니다.

KEEP 버퍼 풀 및 RECYCLE 버퍼 풀은 특별한 버퍼 풀 튜닝에 사용됩니다. KEEP 버퍼 풀은 LRU가 일반적으로 버퍼를 보유하는 것보다 오랫동안 메모리에 버퍼를 보유하도록 설계되었습니다. RECYCLE 버퍼 풀은 일반적으로 LRU가 버퍼를 비우는 것보다 빠르게 버퍼를 메모리에서 비울 수 있도록 설계되었습니다.

기본 블록과 다른 크기의 블록을 보관하도록 추가 버퍼 캐시를 구성할 수 있습니다.

## 리두 로그 버퍼

- SGA의 순환 버퍼임
- 데이터베이스에 대한 변경 사항 관련 정보를 보관함
- DML 및 DDL 등의 작업이 수행한 리두 변경 사항 관련 정보가 들어 있는 리두 항목을 포함함



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 리두 로그 버퍼

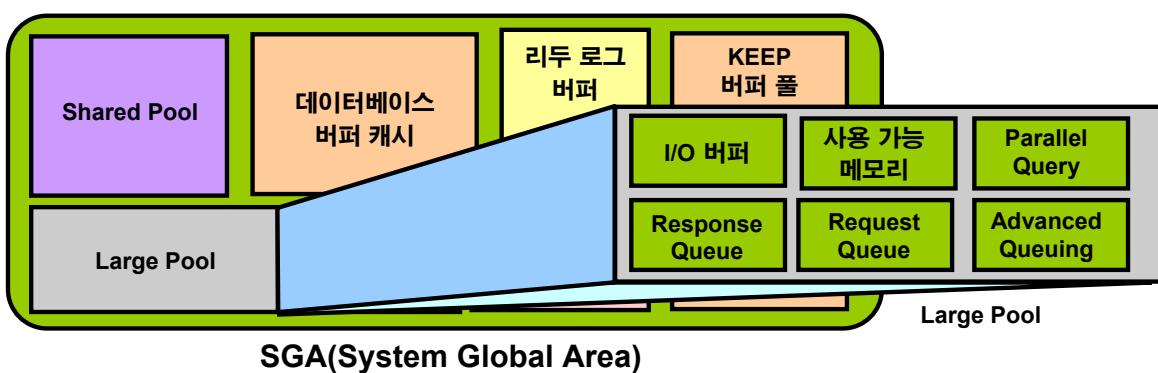
리두 로그 버퍼는 데이터베이스에 대한 변경 사항 관련 정보가 포함된 SGA의 순환 버퍼입니다. 이 정보는 리두 항목에 저장됩니다. 리두 항목은 DML, DDL 또는 내부 작업에 의해 데이터베이스에 수행된 변경 사항을 재생성(또는 리두)하는 데 필요한 정보를 포함합니다. 필요한 경우 리두 항목은 데이터베이스 Recovery에 사용됩니다.

서버 프로세스가 버퍼 캐시에 변경 사항을 적용할 때 리두 항목이 생성되고 SGA의 리두 로그 버퍼에 기록됩니다. 리두 항목은 버퍼에서 연속되는 순차적 공간을 차지합니다. 로그 기록자 백그라운드 프로세스는 리두 로그 버퍼를 디스크의 활성 리두 로그 파일(또는 파일 그룹)에 기록합니다.

# Large Pool

다음을 위한 대규모 메모리 할당을 제공합니다.

- Shared Server 및 Oracle XA 인터페이스용 세션 메모리
- I/O 서버 프로세스
- 오라클 데이터베이스 백업 및 복원 작업



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Large Pool

데이터베이스 관리자는 다음을 위한 대규모 메모리 할당을 제공하기 위해 *Large Pool*이라는 선택적 메모리 영역을 구성할 수 있습니다.

- Shared Server 및 Oracle XA 인터페이스용 세션 메모리(트랜잭션이 여러 데이터베이스와 상호 작용하는 경우 사용됨):
- I/O 서버 프로세스
- 오라클 데이터베이스 백업 및 복원 작업
- Parallel Query 작업
- Advanced Queuing 메모리 테이블 저장 영역

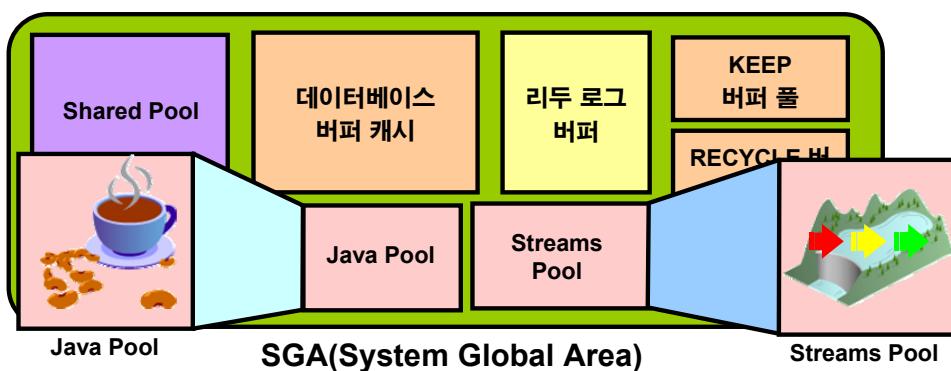
Shared Server, Oracle XA 또는 Parallel Query 버퍼에 대해 Large Pool에서 세션 메모리를 할당함으로써 오라클 데이터베이스는 기본적으로 공유 SQL을 캐시에 저장하는데 Shared Pool을 사용할 수 있으며, 공유 SQL 캐시를 축소(Shrink)하면 발생하는 성능 오버헤드를 방지할 수 있습니다.

또한 오라클 데이터베이스 백업 및 복원 작업, I/O 서버 프로세스 및 병렬 버퍼에 사용되는 메모리는 크기가 수백 KB인 버퍼에서 할당됩니다. Large Pool은 Shared Pool보다 이러한 대규모 메모리 요청을 더 잘 만족시켜 줄 수 있습니다.

Large Pool은 LRU(Least Recently Used) List로 관리되지 않습니다.

# Java Pool 및 Streams Pool

- Java Pool 메모리는 JVM의 모든 세션별 Java 코드 및 데이터를 저장하는 데 사용되며,
- Streams Pool 메모리는 Oracle Streams에서 다음 작업 전용으로 사용됩니다.
  - 버퍼링된 큐 메시지 저장
  - Oracle Streams 프로세스용 메모리 제공



Copyright © 2010, Oracle. All rights reserved.

## Java Pool 및 Streams Pool

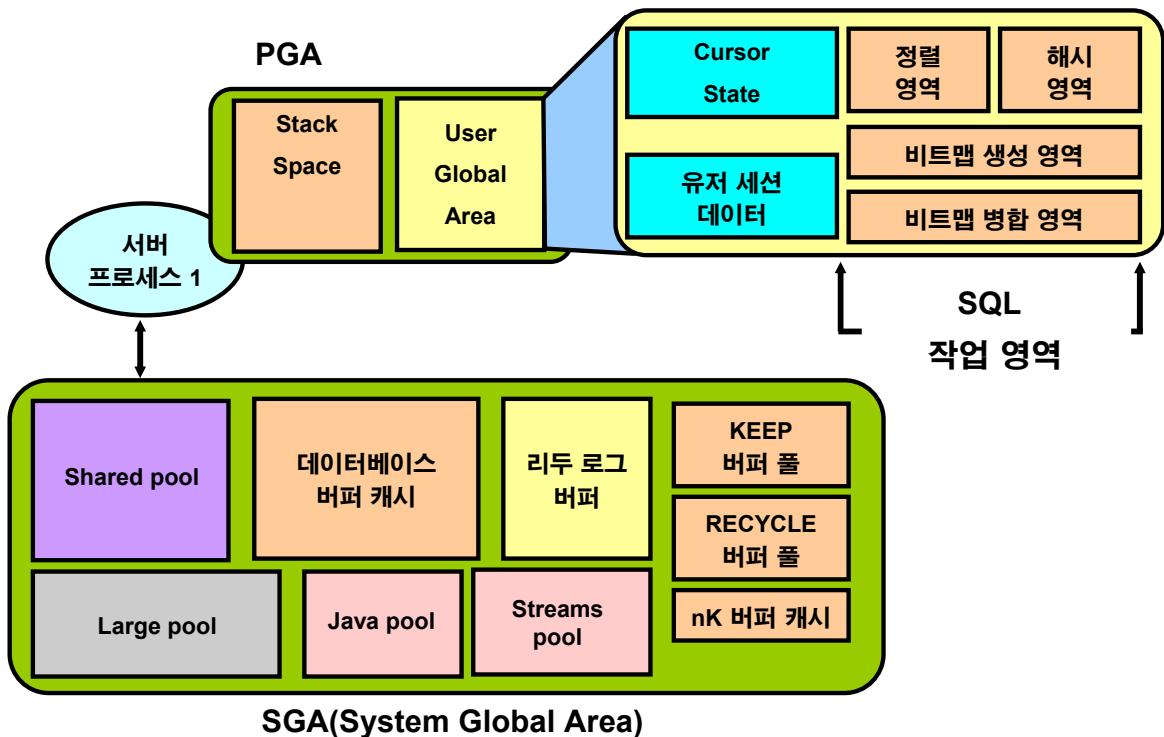
Java Pool 메모리는 JVM의 모든 세션별 Java 코드 및 데이터를 저장하는 데 사용되며, 오라클 데이터베이스를 실행 중인 모드에 따라 다른 방식으로 사용됩니다.

Streams Pool은 Oracle Streams 전용으로 사용되며, 버퍼링된 큐 메시지를 저장하고 Oracle Streams 캡처 프로세스 및 적용 프로세스에 대해 메모리를 제공합니다.

별도로 구성하지 않으면 Streams Pool의 크기는 0에서 시작하며 Oracle Streams가 사용될 때 필요에 따라 동적으로 커집니다.

주: 이 강의에서는 Java 프로그래밍 및 Oracle Streams에 대해 자세히 설명하지 않습니다.

# PGA(Program Global Area)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## PGA(Program Global Area)

PGA(Program Global Area)는 서버 프로세스에 대한 데이터 및 제어 정보를 포함하는 전용(Private) 메모리 영역입니다. 각 서버 프로세스에는 고유한 PGA가 포함됩니다. PGA에 대한 액세스는 해당 서버 프로세스에 배타적이며 대신 작업을 수행하는 Oracle 코드에 의해서만 읽혀집니다. 개발자 코드에서는 사용할 수 없습니다.

모든 PGA에는 Stack Space가 포함됩니다. Dedicated Server 환경에서는 데이터베이스 Instance에 연결하는 각 유저가 별개의 서버 프로세스를 갖습니다. 이 유형의 연결에서 PGA에는 UGA(User Global Area)로 알려진 메모리의 하위 부분이 포함됩니다. UGA는 다음으로 구성됩니다.

- 커서에 대한 런타임 정보를 저장하기 위한 커서 영역
- 세션에 대한 제어 정보를 위한 유저 세션 데이터 저장 영역
- 다음으로 구성되는 SQL 문 처리를 위한 SQL 작업 영역:
  - ORDER BY 및 GROUP BY와 같이 데이터를 정렬하는 함수를 위한 정렬 영역
  - 테이블에 대해 해시 조인을 수행하는 해시 영역
  - 데이터 웨어하우스에 일반적인 비트맵 인덱스 생성에 사용되는 비트맵 생성 영역
  - 비트맵 인덱스 계획 실행을 분석하기 위해 사용되는 비트맵 병합 영역

Shared Server 환경에서는 여러 클라이언트 유저가 서버 프로세스를 공유합니다. 이 모델에서 UGA는 SGA(Shared Pool 또는 Large Pool(구성된 경우))로 이동하여 PGA에는 Stack Space만 남습니다.

## 퀴즈

서버 또는 백그라운드 프로세스의 데이터 및 제어 정보를 포함하는 메모리 영역은 무엇입니까?

- 1. Shared Pool**
- 2. PGA**
- 3. 버퍼 캐시**
- 4. 유저 세션 데이터**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 퀴즈

데이터 파일에서 데이터베이스 버퍼 캐시로 읽혀지는 것은 무엇입니까?

1. 행
2. 변경 사항
3. 블록
4. SQL

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 3

# 프로세스 구조

- **User Process**
  - 오라클 데이터베이스에 연결하는 응용 프로그램 또는 도구
- **데이터베이스 프로세스**
  - 서버 프로세스: Oracle Instance에 연결되면 유저가 세션을 설정하면 시작됩니다.
  - 백그라운드 프로세스: Oracle Instance가 시작될 때 시작됩니다.
- **Daemon/응용 프로그램 프로세스**
  - 네트워킹 리스너
  - 그리드 Infrastructure Daemon

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 프로세스 구조

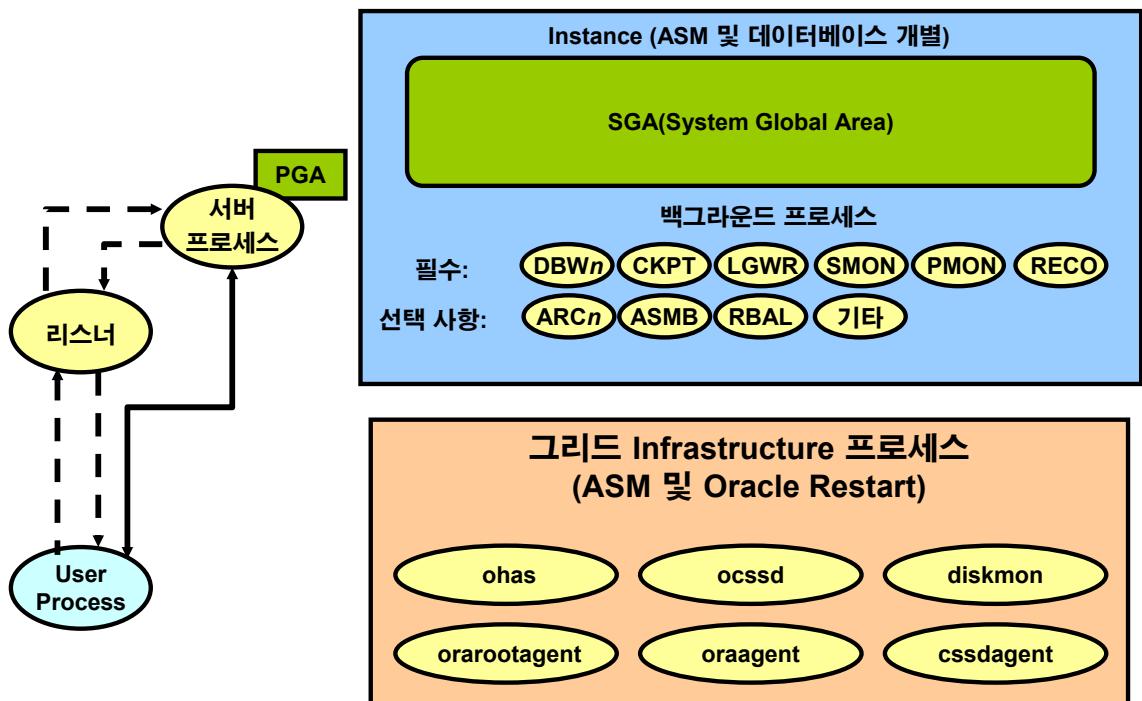
오라클 데이터베이스 시스템의 프로세스는 세 가지 주요 그룹으로 나눌 수 있습니다.

- 응용 프로그램 또는 오라클 도구 코드를 실행하는 User Process
- 오라클 데이터베이스 서버 코드를 실행하는 오라클 데이터베이스 프로세스(서버 프로세스 및 백그라운드 프로세스 포함)
- 단일 데이터베이스에 특정되지 않은 Oracle Daemon 및 응용 프로그램 프로세스

유저가 응용 프로그램이나 SQL\*Plus 등의 오라클 도구를 실행할 때 유저의 응용 프로그램을 *user process*라고 부릅니다. User Process는 데이터베이스 서버 시스템에 존재하거나 존재하지 않을 수 있습니다. 또한 오라클 데이터베이스는 서버 프로세스를 생성하여 User Process에서 발행한 명령을 실행합니다. 아울러 Oracle 서버의 Instance에는 서로 간에는 물론 운영 체제와도 상호 작용하여 메모리 구조를 관리하고, I/O를 비동기 방식으로 수행하여 디스크에 데이터를 쓰고, 기타 필요한 작업을 수행하는 백그라운드 프로세스 세트가 있습니다. 각 오라클 데이터베이스 구성의 프로세스 구조는 운영 체제와 선택한 오라클 데이터베이스 옵션에 따라 서로 다릅니다. 연결된 유저에 대한 코드는 Dedicated Server 또는 Shared Server로 구성할 수 있습니다.

- **Dedicated Server:** 각 세션에 대해 데이터베이스 응용 프로그램은 오라클 데이터베이스 서버 코드를 실행하는 Dedicated Server 프로세스가 제공하는 User Process에 의해 실행됩니다.
- **Shared Server:** 각 연결에 대해 Dedicated Server 프로세스가 필요하지 않습니다. 디스패처가 여러 수신 네트워크 세션 요청을 Shared Server 프로세스 풀로 이동시킵니다. Shared Server 프로세스는 모든 클라이언트 요청을 처리합니다.

## 프로세스 구조



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 프로세스 구조

#### 서버 프로세스

오라클 데이터베이스는 서버 프로세스를 생성하여 Instance에 연결된 User Process의 요청을 처리합니다. User Process는 오라클 데이터베이스에 연결하는 응용 프로그램 또는 도구를 나타냅니다. User Process는 오라클 데이터베이스와 동일한 시스템이 있거나 원격 클라이언트에 존재하며 네트워크를 이용하여 오라클 데이터베이스에 연결할 수 있습니다. User Process는 먼저 전용 환경에서 서버 프로세스를 생성하는 리스너 프로세스와 통신합니다.

각 유저의 응용 프로그램 대신 생성된 서버 프로세스는 다음과 같은 작업을 수행할 수 있습니다.

- 응용 프로그램을 통해 발행된 SQL 문 구문 분석 및 실행
- 디스크의 데이터 파일에서 필요한 데이터 블록을 SGA의 공유 데이터베이스 버퍼로 읽기 (SGA에 블록이 이미 표시되어 있지 않은 경우)
- 응용 프로그램이 정보를 처리할 수 있는 방식으로 결과 반환

#### 백그라운드 프로세스

성능을 최대화하고 많은 유저를 수용하기 위해 다중 처리 오라클 데이터베이스 시스템은 백그라운드 프로세스라는 일부 오라클 데이터베이스 프로세스를 추가로 사용합니다. 하나의 오라클 데이터베이스 Instance에는 여러 개의 백그라운드 프로세스가 있을 수 있습니다.

## 프로세스 구조(계속)

일반적으로 비RAC, 비ASM 환경에서 표시되는 백그라운드 프로세스에는 다음이 포함될 수 있습니다.

- DBW $n$ (데이터베이스 기록자 프로세스)
- LGWR(로그 기록자 프로세스)
- CKPT(체크포인트 프로세스)
- SMON(시스템 모니터 프로세스)
- PMON(프로세스 모니터 프로세스)
- RECO(복구자 프로세스)
- CJQ0(조정자 작업 큐)
- J $nnn$ (작업 슬래이브 프로세스)
- ARC $n$ (아카이버 프로세스)
- QMN $n$ (큐 모니터 프로세스)

RAC 등의 고급 구성에는 다른 백그라운드 프로세스도 있습니다. 백그라운드 프로세스에 대한 자세한 사항은 V\$BGPROCESS 뷰를 참조하십시오.

일부 백그라운드 프로세스는 Instance가 시작될 때 자동으로 생성되지만 다른 일부 프로세스는 필요에 따라 시작됩니다.

다른 프로세스 구조는 단일 데이터베이스에 특정적이지 않으며, 동일 서버의 여러 데이터베이스에서 공유될 수 있습니다. 그리드 Infrastructure 및 네트워킹 프로세스는 이 범주에 포함됩니다.

Linux 및 UNIX 시스템의 Oracle 그리드 Infrastructure 프로세스에는 다음이 포함됩니다.

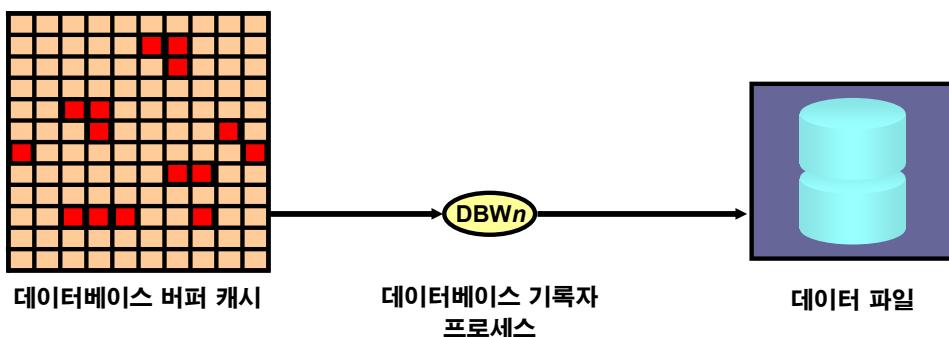
- ohasd: Oracle Clusterware 프로세스를 시작하는 Oracle High Availability Service Daemon
- ocssd: Cluster Synchronization Service Daemon
- diskmon: HP Oracle Exadata Storage Server에 대한 입력 및 출력 한도를 지정하는 Disk Monitor Daemon
- cssdagent: CSS Daemon인 ocssd를 시작 및 정지하고 상태 확인
- oraagent: 오라클 특정 요구 사항 및 복잡한 리소스를 지원하기 위해 클러스터웨어 확장
- orarootagent: 네트워크와 같은 루트 소유 리소스의 관리를 돋는 특별한 오라클 에이전트 프로세스

**주:** 백그라운드 프로세스에 대한 자세한 리스트를 보려면 본 과정의 *오라클 백그라운드 프로세스* 부록 또는 *Oracle Database Reference* 설명서를 참조하십시오.

## DBWn(데이터베이스 기록자 프로세스)

데이터베이스 버퍼 캐시의 수정된(더티) 버퍼를 다음과 같이 디스크에 기록합니다.

- 다른 처리를 수행하는 동안 비동기적으로 기록
- 체크포인트 전진



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### DBWn(데이터베이스 기록자 프로세스)

DBWn(데이터베이스 기록자 프로세스)은 버퍼의 내용을 데이터 파일에 기록합니다. DBWn 프로세스는 데이터베이스 버퍼 캐시의 수정된(더티) 버퍼를 디스크에 기록합니다. 대부분의 시스템에서는 하나의 데이터베이스 기록자 프로세스(DBW0)만 사용하면 되지만, 시스템에서 데이터를 많이 수정하는 경우에는 기록 성능을 개선하기 위해 추가 프로세스(DBW1~DBW9 및 DBWa~DBWz)를 구성할 수 있습니다. 단일 프로세서 시스템에서는 이러한 추가 DBWn 프로세스가 유용하지 않습니다.

데이터베이스 버퍼 캐시의 버퍼는 수정되면 더티로 표시되며 SCN 순서로 보관되는 상위 체크포인트 큐에 추가됩니다. 그러므로 이 SCN 순서는 이들 변경된 버퍼에 대해 리두 로그에 기록되는 리두 순서와 일치합니다. 버퍼 캐시의 사용 가능한 버퍼 수가 내부 임계값(서버 프로세스가 사용 가능한 버퍼를 얻기가 어려운 범위)까지 떨어지면 DBWn은 프로세스가 필요할 때 버퍼를 바꿀 수 있도록 LRU List의 아래쪽에 있는 데이터 파일에 대해 자주 사용되지 않는 버퍼를 기록합니다. DBWn은 또한 체크포인트가 계속 전진할 수 있도록 체크포인트 큐 아래쪽에 버퍼를 기록합니다.

## DBWn(데이터베이스 기록자 프로세스)(계속)

SGA에는 Instance Failure의 경우 Recovery가 시작되어야 하는 리두 스트림 위치의 RBA(리두 바이트 주소)를 포함하는 메모리 구조가 들어 있습니다. 이 구조는 리두에 대한 포인터 역할을 하며, 3초에 한 번씩 CKPT 프로세스에 의해 콘트롤 파일에 기록됩니다. DBWn은 더티 버퍼를 SCN 순서로 기록하며 리두는 SCN 순서로 되어 있으므로 DBWn이 LRUW list에서 더티 버퍼를 기록할 때마다 SGA 메모리 구조에 포함된 포인터도 전진하여 해당 Instance Recovery (필요한 경우)가 정확한 위치에서부터 리두 읽기를 시작하고 불필요한 I/O를 읽지 않도록 합니다. 이를 *Incremental 체크포인트*라고 합니다.

**주:** DBWn은 테이블스페이스가 읽기 전용으로 설정되거나 오프라인으로 전환되는 등의 경우에도 기록을 수행할 수 있습니다. 이러한 경우에는 해당하는 데이터 파일에만 속하는 더티 버퍼가 SCN 순서와 관계없이 데이터베이스에 기록되므로 Incremental 체크포인트는 발생하지 않습니다.

LRU 알고리즘은 디스크 읽기를 최소화하기 위해 보다 자주 액세스되는 블록을 버퍼 캐시에 유지합니다. CACHE 옵션을 테이블에 적용하면 블록을 메모리에서 더 오랫동안 보존할 수 있습니다.

DB\_WRITER\_PROCESSES 초기화 파라미터는 DBWn 프로세스 수를 지정합니다. 최대 DBWn 프로세스 수는 36개입니다. 시작 중에 유저가 이 수를 지정하지 않으면 오라클 데이터베이스가 CPU 및 프로세스 그룹의 수에 따라 DB\_WRITER\_PROCESSES를 설정하는 방법을 결정합니다.

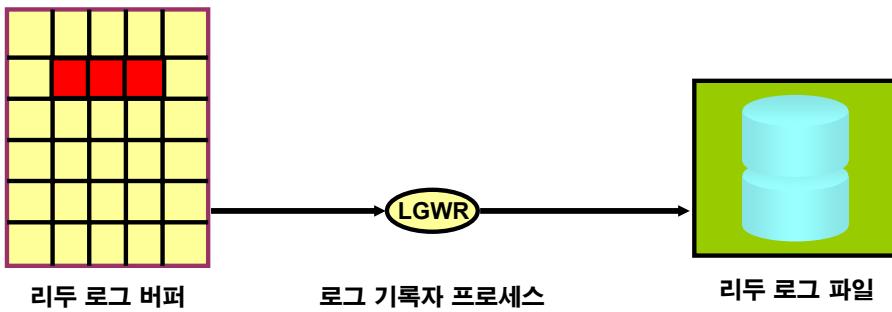
DBWn 프로세스는 다음과 같은 경우에 더티 버퍼를 디스크에 기록합니다.

- 서버 프로세스가 버퍼를 임계값 수만큼 스캔한 후에 재사용 가능한 클린 버퍼를 찾지 못하면 DBWn에 기록을 수행하라는 신호를 보냅니다. 그러면 DBWn은 다른 처리를 수행하면서 더티 버퍼를 디스크에 비동기 방식으로 기록합니다.
- DBWn은 버퍼를 기록하여 Instance Recovery가 시작되는 리두 스티드(로그)의 위치인 체크포인트를 전진시킵니다. 이 로그 위치는 버퍼 캐시의 가장 오래된 더티 버퍼에 의해 결정됩니다.

어떤 경우든 DBWn은 효율성을 개선하기 위해 일괄 처리된(다중 블록) 쓰기를 수행합니다. 다중 블록 쓰기에서 기록되는 블록 수는 운영 체제에 따라 다릅니다.

## LGWR(로그 기록자 프로세스)

- 리두 로그 버퍼를 디스크의 리두 로그 파일에 기록합니다.
- 기록은 다음과 같은 경우에 수행됩니다.
  - User Process가 트랜잭션을 커밋할 때
  - 리두 로그 버퍼가 1/3 찼을 때
  - DBWn 프로세스가 수정된 버퍼를 디스크에 기록하기 전에
  - 3초마다



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### LGWR(로그 기록자 프로세스)

LGWR(로그 기록자 프로세스)은 리두 로그 버퍼 항목을 디스크의 리두 로그 파일에 기록하여 리두 로그 버퍼를 관리하며, 마지막으로 기록된 이후 버퍼로 복사된 모든 리두 항목을 기록합니다.

리두 로그 버퍼는 일종의 순환 버퍼이므로 LGWR가 리두 로그 버퍼의 리두 항목을 리두 로그 파일에 기록하면 서버 프로세스가 새 항목을 복사하여 디스크에 기록된 리두 로그 버퍼의 항목을 덮쳐쓸 수 있습니다. LGWR는 일반적으로 리두 로그에 대한 액세스가 많을 때도 새 항목을 위한 버퍼 공간이 항상 존재할 수 있을 정도로 빠르게 쓰기를 수행합니다. 또한 연속하는 한 개의 버퍼 부분을 디스크에 기록합니다.

LGWR는 다음 경우 기록합니다.

- User Process가 트랜잭션을 커밋할 때
- 리두 로그 버퍼가 1/3 찼을 때
- DBWn 프로세스가 수정된 버퍼를 디스크에 기록하기 전에(필요한 경우)
- 3초마다

## LGWR(LogWriter 프로세스) (계속)

DBW $n$ 이 수정된 버퍼를 기록하려면 버퍼에 대한 변경 사항과 연관된 모든 리두 레코드를 디스크에 기록해야 합니다(먼저 쓰기 프로토콜). DBW $n$ 은 일부 리두 레코드가 기록되지 않았음을 확인하면 LGWR에 리두 레코드를 디스크에 기록하라는 신호를 보냅니다. LGWR가 리두 로그 버퍼 기록을 완료해야 DBW $n$ 이 데이터 버퍼를 기록할 수 있습니다. LGWR는 현재 로그 그룹에 레코드를 기록합니다. 해당 그룹의 파일 중 하나가 손상되거나 사용할 수 없는 상태이면 LGWR는 그룹의 다른 파일에 기록을 계속하고 오류를 LGWR Trace File 및 시스템 Alert Log에 기록합니다. 그룹의 모든 파일이 손상되었거나 그룹을 아카이브하지 않아 사용할 수 없는 경우에는 LGWR가 계속해서 작동할 수 없습니다.

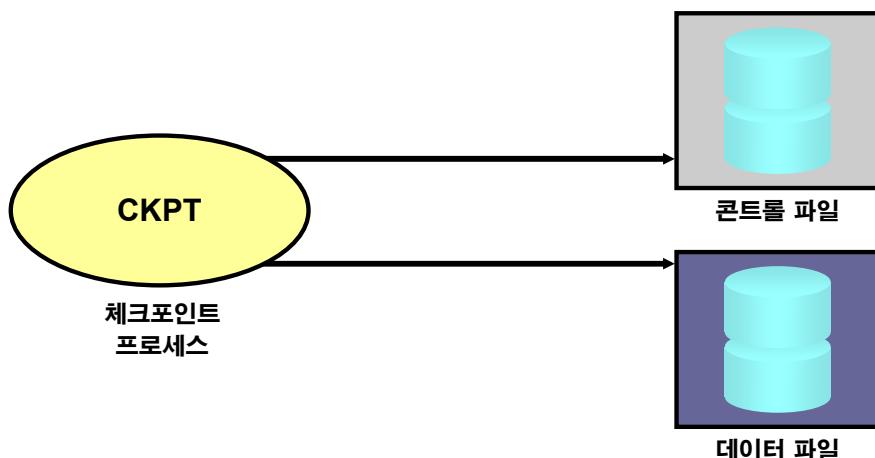
유저가 COMMIT 문을 실행하면 LGWR는 리두 로그 버퍼에 커밋 레코드를 배치하고 이를 트랜잭션의 리두 항목과 함께 디스크에 즉시 기록합니다. 데이터 블록에 대한 해당하는 변경 사항은 보다 효율적으로 기록될 수 있을 때까지 지연됩니다. 이를 빠른 커밋 메커니즘(*fast commit mechanism*)이라고 합니다. 트랜잭션의 커밋 레코드를 포함하는 리두 항목의 원자 쓰기는 트랜잭션의 커밋 여부를 확인하는 단일 이벤트입니다. 오라클 데이터베이스는 데이터 버퍼를 아직 디스크에 기록하지 않은 경우에도 커밋 트랜잭션에 대해 성공 코드를 반환합니다.

버퍼 공간이 더 필요한 경우 LGWR는 트랜잭션이 커밋되기 전에 리두 로그 항목을 기록하기도 합니다. 이러한 항목은 나중에 트랜잭션이 커밋되는 경우에만 영구적으로 보존됩니다. 유저가 트랜잭션을 커밋하면 트랜잭션에 SCN(시스템 변경 번호)이 할당되는데, 오라클 데이터베이스는 이 번호를 트랜잭션의 리두 항목과 함께 리두 로그에 기록합니다. SCN은 Recovery 작업을 Real Application Clusters 및 분산 데이터베이스에서 동기화할 수 있도록 하기 위해 리두 로그에 기록됩니다.

작업량이 많을 때 LGWR는 그룹 커밋을 사용하여 리두 로그 파일에 기록할 수 있습니다. 예를 들어, 유저가 트랜잭션을 커밋한다고 가정해 보겠습니다. 이 경우 LGWR는 해당 트랜잭션의 리두 항목을 디스크에 기록해야 합니다. 그러면 다른 유저는 COMMIT 문을 실행합니다. 그러나 LGWR는 이전 쓰기 작업을 완료하기 전까지는 이러한 트랜잭션을 커밋하기 위해 리두 로그 파일에 기록할 수 없습니다. 첫번째 트랜잭션의 항목이 리두 로그 파일에 기록되면 대기 트랜잭션(아직 커밋되지 않은 트랜잭션)의 전체 리두 항목 리스트를 한 번의 작업으로 디스크에 기록할 수 있으므로 트랜잭션 항목을 개별적으로 처리하는 것에 비해 필요한 I/O를 줄일 수 있습니다. 이를 통해 오라클 데이터베이스는 디스크 I/O를 최소화하고 LGWR의 성능을 최대화합니다. 커밋에 대한 요청의 속도가 계속 높게 유지되면 리두 로그 버퍼에서 LGWR가 수행하는 모든 쓰기에 여러 커밋 레코드가 포함될 수 있습니다.

## CKPT(체크포인트 프로세스)

- 체크포인트 정보를 기록하는 위치:
  - 콘트롤 파일
  - 각 데이터 파일 헤더



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

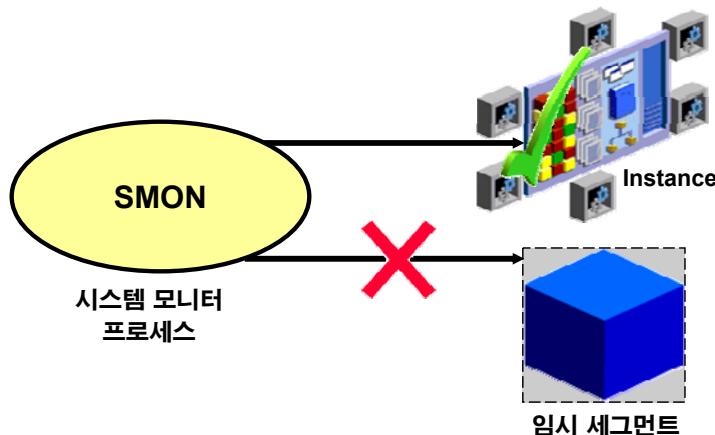
### CKPT(체크포인트 프로세스)

체크포인트는 데이터베이스의 리두 슬레드에 SCN(시스템 변경 번호)을 정의하는 데이터 구조로, 콘트롤 파일 및 각 데이터 파일 헤더에 기록되는 Recovery의 필수 요소입니다.

체크포인트가 발생할 경우 오라클 데이터베이스는 모든 데이터 파일의 헤더를 갱신하여 체크포인트의 세부 사항을 기록해야 하는데, 이 작업은 CKPT 프로세스에 의해 수행됩니다. CKPT 프로세스는 블록을 디스크에 기록하지 않지만 DBWn은 항상 블록을 디스크에 기록합니다. 파일 헤더에 기록된 SCN은 해당 SCN이 디스크에 기록되기 전에 데이터베이스 블록에 대한 모든 변경이 이루어졌음을 나타냅니다.

## SMON(시스템 모니터 프로세스)

- Instance 시작 시 Recovery 수행
- 사용하지 않는 임시 세그먼트 정리



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

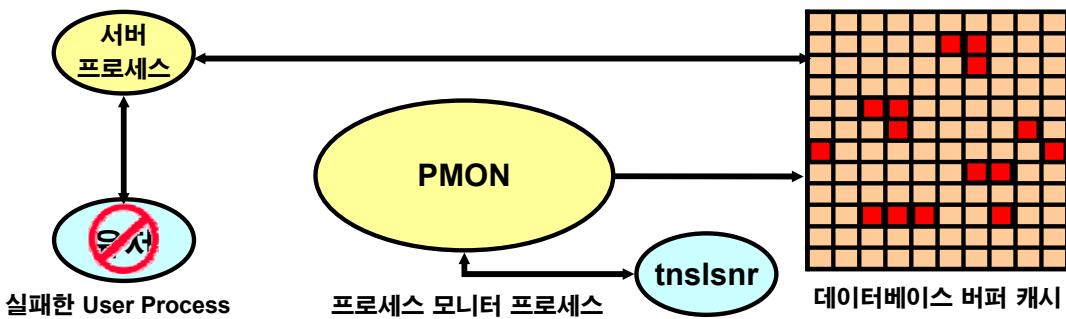
### SMON(시스템 모니터 프로세스)

SMON(시스템 모니터 프로세스)은 필요한 경우 Instance가 시작될 때 Recovery를 수행하며, 더 이상 사용하지 않는 임시 세그먼트도 정리합니다. 파일 읽기 또는 오프라인 오류로 인해 Instance Recovery 도중 종료된 트랜잭션을 건너뛰는 경우 테이블스페이스나 파일이 다시 온라인으로 전환되면 SMON이 이를 Recovery합니다.

SMON은 프로세스가 필요한지 여부를 정기적으로 확인합니다. SMON이 필요할 경우 다른 프로세스에서 SMON을 호출할 수 있습니다.

## PMON(프로세스 모니터 프로세스)

- User Process가 실패할 경우 프로세스 Recovery 수행
  - 데이터베이스 버퍼 캐시 정리
  - User Process에서 사용하는 리소스 해제
- Idle 세션 타임아웃에 대한 세션 모니터
- 리스너에 동적으로 데이터베이스 서비스 등록



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### PMON(프로세스 모니터 프로세스)

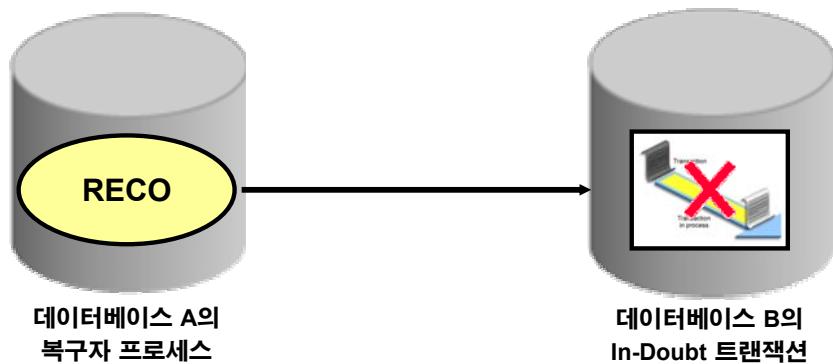
PMON(프로세스 모니터 프로세스)은 User Process가 실패할 경우 프로세스 Recovery를 수행하고, 데이터베이스 버퍼 캐시를 정리하며, User Process에서 사용하던 리소스를 해제합니다. 예를 들어, PMON은 활성 트랜잭션 테이블의 상태를 재설정하고 Lock을 해제하여 활성 프로세스 리스트에서 프로세스 ID를 제거합니다.

PMON은 정기적으로 디스패처와 서버 프로세스의 상태를 확인하여 실행이 정지된 경우(오라클 데이터베이스에서 의도적으로 종료한 경우 제외) 이를 재시작합니다. 또한 Instance 및 디스패처 프로세스에 대한 정보도 네트워크 리스너에 등록합니다.

SMON과 마찬가지로 PMON은 정기적으로 필요 여부를 확인하여 다른 프로세스에서 PMON을 필요로 할 경우 호출할 수 있습니다.

## 복구자 프로세스

- 분산 데이터베이스 구성에 사용
- In-Doubt 분산 트랜잭션과 관련된 다른 데이터베이스에 자동으로 연결
- 모든 In-Doubt 트랜잭션을 자동으로 해결
- In-Doubt 트랜잭션에 해당하는 모든 행 제거



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### RECO(복구자 프로세스)

RECO(복구자 프로세스)는 분산 트랜잭션과 관련된 Failure를 자동으로 해결하는 분산 데이터베이스 구성에 사용되는 백그라운드 프로세스입니다. Instance의 RECO 프로세스는 In-Doubt 분산 트랜잭션과 관련된 다른 데이터베이스에 자동으로 연결됩니다. RECO 프로세스는 관련 데이터베이스 서버 간의 연결을 재설정할 때 모든 In-Doubt 트랜잭션을 자동으로 해결하여 각 데이터베이스의 보류 트랜잭션 테이블에서 해결된 In-Doubt 트랜잭션에 해당하는 모든 행을 제거합니다.

RECO 프로세스가 원격 서버와 연결하지 못하면 RECO는 지정된 간격 후에 자동으로 연결을 재시도합니다. 그러나 시간이 지남에 따라 RECO는 다른 연결을 시도하기 전까지 보다 오랜 시간 동안 기다립니다(시간이 지수적으로 증가함).

## ARCn(아카이버 프로세스)

- 로그 스위치가 발생한 후에 리두 로그 파일을 지정된 기억 장치로 복사합니다.
- 트랜잭션 리두 데이터를 수집하여 대기 대상으로 전송할 수 있습니다.



**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

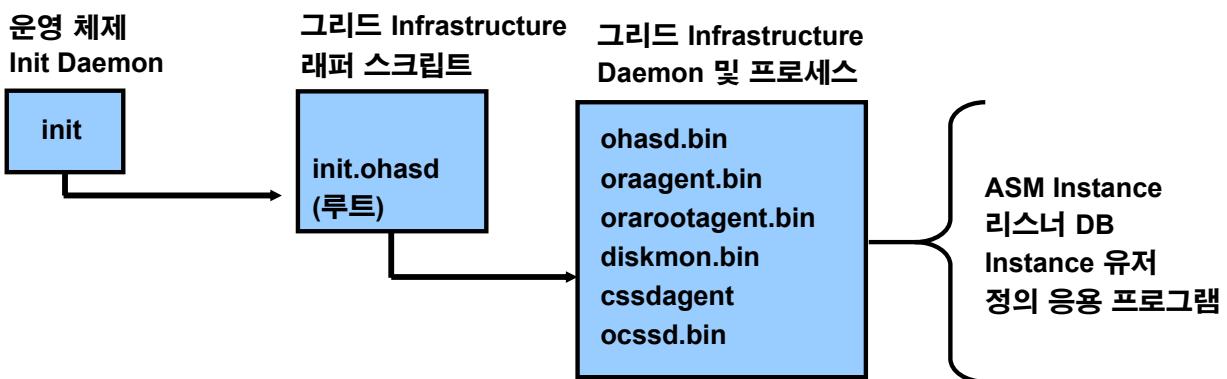
### ARCn(아카이버 프로세스)

ARCn(아카이버 프로세스)은 로그 스위치가 발생한 후에 리두 로그 파일을 지정된 저장 장치로 복사합니다. ARCn 프로세스는 데이터베이스가 ARCHIVELOG 모드이며 자동 아카이브가 활성화되어 있을 때만 표시됩니다.

데이터 대량 로드 시와 같이 아카이브 할 작업 로드가 많을 것으로 예상되면 아카이버 프로세스의 최대 수를 늘릴 수 있습니다. 아카이브 로그 대상은 여러 개가 존재할 수 있습니다. 각 대상에 대해 최소한 하나의 아카이버 프로세스를 두는 것이 좋습니다. 아카이버 프로세스의 기본값은 4개입니다.

## 프로세스 시작 시퀀스

- Oracle 그리드 Infrastructure는 OS init Daemon에 의해 시작됩니다.



- Oracle 그리드 Infrastructure를 설치하면 시스템이 시작할 때마다 해당하는 실행 레벨로 시작되도록 보장하기 위해 /etc/inittab 파일이 수정됩니다.

```
# cat /etc/inittab
..
h1:35:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 프로세스 시작 시퀀스

Oracle 그리드 Infrastructure를 설치하면 래퍼 스크립트를 시작하기 위한 항목들이 운영 체제의 /etc/inittab 파일에 배치됩니다. 래퍼 스크립트는 환경 변수를 설정한 다음 Oracle 그리드 Infrastructure Daemon과 프로세스를 시작합니다.

명령을 사용하여 Oracle 그리드 Infrastructure를 정지하면 Daemon이 정지되지만 래퍼 스크립트 프로세스는 계속 실행됩니다.

UNIX /etc/inittab 파일의 형식은 다음과 같습니다.

id : run levels : action : process with parameters

래퍼 스크립트는 종료될 때마다 다시 시작되도록 재생성된 작업으로 실행됩니다.

일부 Oracle 그리드 Infrastructure Daemon은 실시간 우선 순위에 따라 Root 유저로 실행되며, 그 외 Daemon은 시작된 후 유저 모드 우선 순위에 따라 그리드 Infrastructure 소유자로 실행됩니다. Windows 플랫폼에서 운영 체제 서비스는 래퍼 초기화 스크립트 대신 사용되며 daemon은 실행 Binary File입니다.

주: 래퍼 스크립트는 직접 실행할 수 없습니다.

# 데이터베이스 저장 영역 구조



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 저장 영역 구조

오라클 데이터베이스는 다음과 같은 파일로 구성되어 있습니다.

- 콘트롤 파일:** 데이터베이스 자체에 대한 데이터(즉, 물리적 데이터베이스 구조 정보)를 포함합니다. 이 파일은 데이터베이스에 매우 중요합니다. 이 파일이 없으면 데이터베이스 내의 데이터에 액세스할 때 데이터 파일을 열 수 없습니다. 이 파일은 또한 백업과 관련된 메타 데이터를 포함할 수 있습니다.
- 데이터 파일:** 데이터베이스의 유저 또는 응용 프로그램 데이터, 메타 데이터 및 데이터 딕셔너리를 포함합니다.
- 온라인 리두 파일:** 데이터베이스의 Instance Recovery를 허용합니다. 데이터베이스 서버가 손상되었지만 해당 데이터 파일은 손실되지 않은 경우 Instance는 이러한 파일 안에 있는 정보를 사용하여 데이터베이스를 Recovery할 수 있습니다.

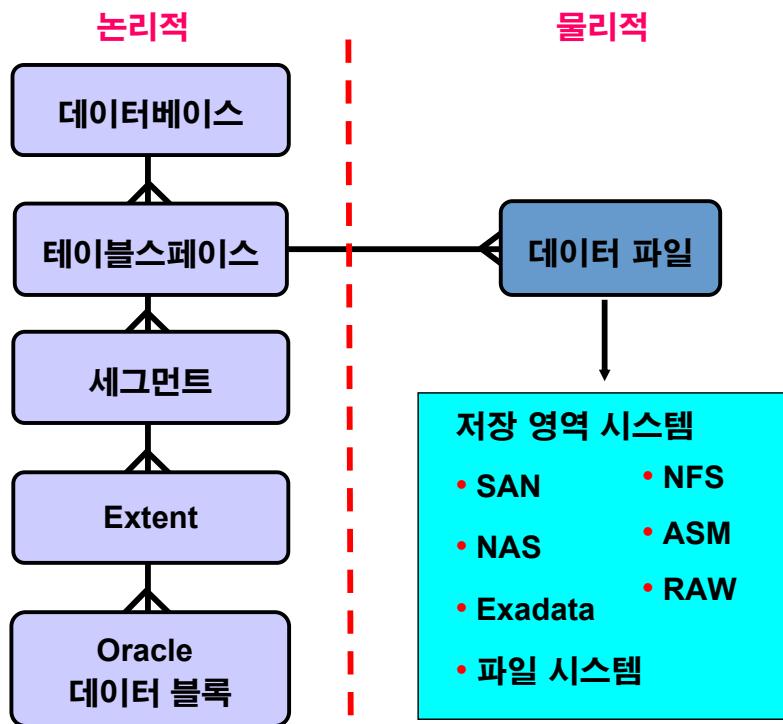
다음은 성공적인 데이터베이스 실행에 필요한 추가 파일입니다.

- 파라미터 파일:** Instance 시작 시 어떻게 Instance를 구성할지 정의하는 데 사용됩니다.
- Password File:** sysdba, sysoper 및 sysasm 롤이 Instance에 원격으로 연결하여 관리 작업을 수행할 수 있도록 합니다.
- 백업 파일:** 데이터베이스 Recovery에 사용됩니다. 백업 파일은 일반적으로 Media Failure 또는 User Error로 원본 파일이 손상되었거나 삭제되었을 경우에 복원합니다.
- 아카이브된 리두 파일:** Instance에 의해 생성되는 데이터 변경(리두)에 대한 기록을 지속적으로 포함합니다. 이 파일과 데이터베이스 백업을 사용하면 손실된 데이터 파일을 Recovery할 수 있습니다. 즉, 아카이브 로그는 복원된 데이터 파일의 Recovery를 가능하게 합니다.

## 데이터베이스 저장 영역 구조(계속)

- **Trace file:** 각 서버와 백그라운드 프로세스는 연관된 Trace File에 정보를 기록할 수 있습니다. 내부 오류가 프로세스에서 감지되면 프로세스는 오류에 대한 정보를 해당 Trace File에 덤프합니다. Trace File에 기록된 정보 중 일부는 데이터베이스 관리자가 사용하고 일부는 오라클 고객 지원 센터에서 사용하게 됩니다.
  - **Alert Log File:** 특수 Trace 항목으로, 데이터베이스의 Alert Log에는 메시지와 오류가 시간 순으로 기록되어 있습니다. Oracle은 Alert Log를 정기적으로 검토할 것을 권장합니다.
- 주: 파라미터 파일, Password File, Alert File 및 Trace File에 대해서는 다른 단원에서 설명합니다.

# 논리적 및 물리적 데이터베이스 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 논리적 및 물리적 데이터베이스 구조

데이터베이스에는 논리적 구조와 물리적 구조가 있습니다.

### 데이터베이스, 테이블스페이스 및 데이터 파일

위의 슬라이드에 데이터베이스, 테이블스페이스 및 데이터 파일 간의 관계가 나타나 있습니다. 각각의 데이터베이스는 논리적으로 두 개 이상의 테이블스페이스로 나뉩니다. 각각의 테이블스페이스가 테이블스페이스에 있는 모든 세그먼트의 데이터를 물리적으로 저장할 수 있도록 하나 이상의 데이터 파일이 명시적으로 생성됩니다. TEMPORARY 테이블스페이스인 경우 데이터 파일 대신 임시 파일이 포함됩니다. 테이블스페이스의 데이터 파일은 지원되는 모든 저장 기술을 사용하여 물리적으로 저장할 수 있습니다.

### 테이블스페이스

데이터베이스는 테이블스페이스라는 논리적 저장 영역 단위로 나뉘어 관련된 논리적 구조 또는 데이터 파일을 함께 그룹화합니다. 예를 들어, 테이블스페이스는 일반적으로 응용 프로그램의 모든 세그먼트를 간단히 몇 개의 관리 작업으로 그룹화합니다.

### 데이터 블록

가장 작은 세분성 레벨에서 오라클 데이터베이스의 데이터는 데이터 블록에 저장됩니다. 하나의 데이터 블록은 디스크에서 특정 바이트 수의 물리적 공간에 해당합니다. 데이터 블록 크기는 생성 시 각 테이블스페이스에 대해 지정됩니다. 데이터베이스는 사용 가능한 데이터베이스 공간을 Oracle 데이터 블록으로 사용 및 할당합니다.

## 논리적 및 물리적 데이터베이스 구조(계속)

### Extent

그 다음 레벨의 논리적 데이터베이스 공간은 *extent*입니다. Extent는 단일 할당으로 얻은 일정 수의 연속적인 오라클 데이터 블록으로, 특정 유형의 정보를 저장하는 데 사용됩니다. Extent에 있는 오라클 데이터 블록은 논리적으로 연속된 데이터이지만 RAID 스트라이핑 및 파일 시스템 구현으로 인해 물리적으로는 디스크에 분산되어 있을 수 있습니다.

### 세그먼트

Extent 다음 레벨의 논리적 데이터베이스 저장 영역은 세그먼트입니다. 세그먼트는 특정 논리적 구조에 할당된 일련의 Extent입니다. 예:

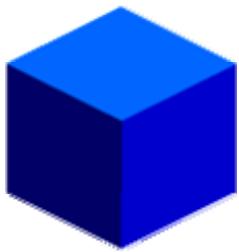
- **데이터 세그먼트:** 클러스터화되지 않은 각각의 비인덱스 구성(Non-index-organized) 테이블에는 데이터 세그먼트가 있습니다. 단, External Table, Global 임시 테이블(Temporary Table), Partition 테이블은 예외입니다. 이 경우 각 테이블에는 한 개 이상의 세그먼트가 있습니다. 테이블 데이터는 모두 해당 데이터 세그먼트의 Extent에 저장됩니다. Partition 테이블의 경우 각 Partition은 데이터 세그먼트를 가집니다. 각 클러스터는 데이터 세그먼트를 가집니다. 클러스터에 있는 모든 테이블의 데이터는 클러스터의 데이터 세그먼트에 저장됩니다.
- **인덱스 세그먼트:** 각 인덱스는 해당 데이터를 모두 저장하는 인덱스 세그먼트를 가집니다. Partition 인덱스의 경우 각 Partition은 인덱스 세그먼트를 가집니다.
- **언두 세그먼트:** 각 데이터베이스 Instance당 하나의 UNDO 테이블스페이스가 생성됩니다. 이 테이블스페이스에는 언두 정보를 임시로 저장하기 위한 다수의 언두 세그먼트가 포함되어 있습니다. 언두 세그먼트의 정보는 데이터베이스 Recovery 중 유저에게 커밋되지 않은 트랜잭션을 롤백하기 위해 읽기 일관성 데이터베이스 정보를 생성하는 데 사용됩니다.
- **임시 세그먼트:** 임시 세그먼트는 SQL 문에서 실행을 완료할 임시 작업 영역이 필요할 때 오라클 데이터베이스에 의해 생성됩니다. 명령문의 실행이 완료되면 나중에 사용할 수 있도록 임시 세그먼트의 Extent가 Instance로 반환됩니다. 모든 유저에 대해 기본 임시 테이블스페이스를 지정하거나 데이터베이스 차원에서 사용할 기본 임시 테이블스페이스를 지정하십시오.

**주:** 위에 나열되지 않은 다른 유형의 세그먼트도 있습니다. 또한 데이터베이스 객체이지만 세그먼트로 간주되지 않는 뷰, 패키지, 트리거 등의 스키마 객체도 있습니다. 세그먼트는 각각 해당하는 디스크 공간 할당을 소유합니다. 다른 객체는 시스템 메타 데이터 세그먼트에 저장된 행으로 존재합니다.

오라클 데이터베이스 서버는 공간을 동적으로 할당합니다. 세그먼트의 기존 Extent가 가득 차면 다른 Extent가 추가됩니다. Extent는 필요에 따라 할당되므로 세그먼트의 Extent는 디스크에서 연속되거나 연속되지 않을 수 있으며, 동일 테이블스페이스에 속하는 다른 데이터 파일에서 올 수도 있습니다.

## 세그먼트, Extent 및 블록

- 세그먼트는 테이블스페이스 내에 존재합니다.
- 세그먼트는 Extent 모음입니다.
- Extent는 데이터 블록 모음입니다.
- 데이터 블록은 디스크 블록에 매핑됩니다.



세그먼트



Extent



데이터 블록

디스크 블록  
(파일 시스템 저장 영역)

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 세그먼트, Extent 및 블록

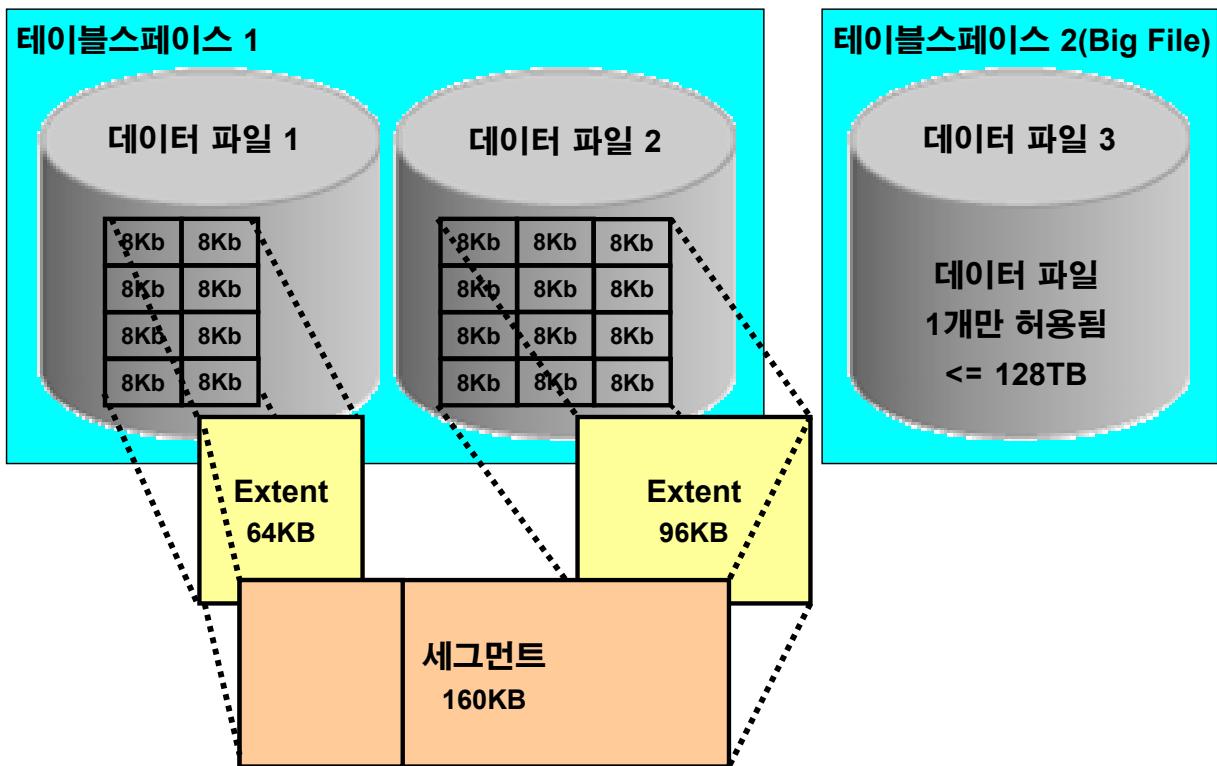
테이블 및 인덱스와 같은 데이터베이스 객체의 하위 집합은 테이블스페이스에 세그먼트로 저장됩니다. 각 세그먼트에는 하나 이상의 Extent가 포함되어 있습니다. Extent는 연속적인 데이터 블록으로 구성됩니다. 즉, 각 Extent는 하나의 데이터 파일에만 존재할 수 있습니다. 데이터 블록은 데이터베이스의 가장 작은 I/O 단위입니다.

데이터베이스에서 OS(운영 체제)의 데이터 블록 집합을 요청하면 OS는 이를 저장 장치의 디스크 블록이나 실제 파일 시스템으로 매핑합니다. 따라서 데이터베이스에 있는 데이터의 물리적 주소를 알 필요가 없으며 데이터 파일을 여러 디스크에 스트라이핑 또는 Mirroring 할 수도 있습니다.

데이터 블록의 크기는 데이터베이스 생성 시 설정할 수 있습니다. 대부분의 데이터베이스의 경우 기본 크기인 8KB가 적합합니다. 큰 테이블 및 인덱스를 사용하는 데이터 웨어하우스 응용 프로그램을 지원하는 데이터베이스인 경우에는 블록 크기를 크게 지정하는 것이 효과적일 수 있습니다.

읽기와 쓰기가 무작위로 발생하는 트랜잭션 응용 프로그램을 지원하는 데이터베이스인 경우에는 블록 크기를 작게 지정하는 것이 유용할 수도 있습니다. 최대 블록 크기는 OS에 따라 다릅니다. 최소 Oracle 블록 크기는 2KB인데 이렇게 작은 크기는 거의 사용되지 않습니다. 테이블스페이스의 블록 크기를 표준이 아닌 다른 값으로 설정할 수 있습니다. 자세한 사항은 *Oracle Database Administrator's Guide*를 참조하십시오.

## 테이블스페이스 및 데이터 파일



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 테이블스페이스 및 데이터 파일

데이터베이스는 관련된 논리적 구조를 그룹화하는 데 사용할 수 있는 테이블스페이스라는 논리적 저장 영역 단위로 나뉩니다. 각 데이터베이스는 논리적으로 SYSTEM 및 SYSAUX의 두 개 이상의 테이블스페이스로 나뉩니다. 각 테이블스페이스에는 테이블스페이스에 있는 모든 논리적 구조 데이터를 물리적으로 저장할 수 있는 하나 이상의 데이터 파일이 명시적으로 생성됩니다.

슬라이드의 그래픽은 두 개의 데이터 파일로 구성된 테이블스페이스 1을 보여 줍니다. 160KB 크기의 세그먼트는 두 개의 Extent로 구성된 두 개의 데이터 파일에 걸쳐 있습니다. 크기가 64KB인 첫번째 Extent는 첫번째 데이터 파일에 있고 크기가 96KB인 두번째 Extent는 두번째 데이터 파일에 있습니다. 두 Extent는 모두 연속된 8Kb Oracle 블록으로 형성됩니다.

주: 매우 큰 파일이 하나만 포함된 Big File 테이블스페이스를 생성할 수도 있습니다. 이 파일의 크기는 Row ID 구조가 허용하는 최대 크기까지 설정할 수 있습니다. 최대 크기는 테이블스페이스의 블록 크기에  $2^{36}$ 을 곱한 값(32KB 블록 크기의 경우 128TB)입니다. 기존의 Small File 테이블스페이스(기본값)는 여러 데이터 파일을 포함할 수 있지만 Big File 테이블스페이스만큼 큰 파일은 포함할 수 없습니다. Big File 테이블스페이스에 대한 자세한 사항은 *Oracle Database Administrator's Guide*를 참조하십시오.

## SYSTEM 및 SYSAUX 테이블스페이스

- **SYSTEM 및 SYSAUX 테이블스페이스는 데이터베이스 생성 시에 생성되는 필수 테이블스페이스로, 온라인 상태여야 합니다.**
- **SYSTEM 테이블스페이스는 데이터 딕셔너리 테이블과 같은 핵심 기능을 위해 사용됩니다.**
- **보조 테이블스페이스인 SYSAUX는 Enterprise Manager Repository 등 추가적인 데이터베이스 구성 요소에 사용됩니다.**
- **SYSTEM 및 SYSAUX 테이블스페이스는 응용 프로그램의 데이터를 저장하는 데 사용하지 않는 것이 좋습니다.**

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### SYSTEM 및 SYSAUX 테이블스페이스

각 오라클 데이터베이스에는 SYSTEM 테이블스페이스 및 SYSAUX 테이블스페이스가 포함되어야 합니다. 이 테이블스페이스는 데이터베이스가 생성될 때 자동으로 생성됩니다. 시스템 기본값은 Small File 테이블스페이스를 생성하는 것입니다. Big File 테이블스페이스도 생성할 수 있는데 이 테이블스페이스를 사용하면 초대용량 파일을 관리할 수 있습니다.

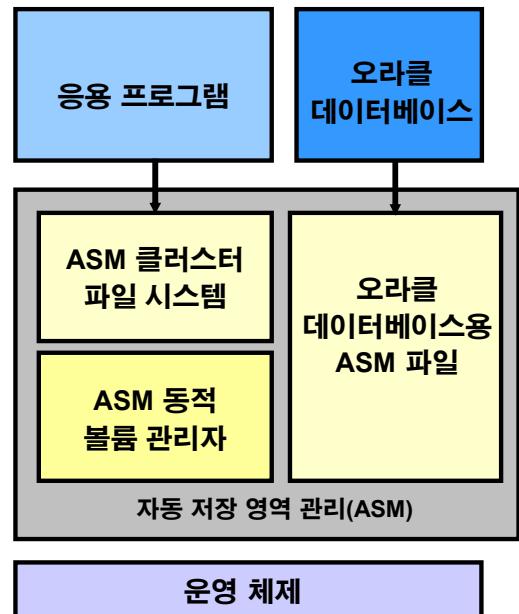
테이블스페이스는 온라인(액세스 가능) 상태이거나 오프라인(액세스 불가) 상태일 수 있습니다. 데이터베이스가 열려 있는 경우 SYSTEM 테이블스페이스는 항상 온라인 상태입니다. SYSTEM 테이블스페이스는 데이터 딕셔너리 테이블 등 데이터베이스의 핵심 기능을 지원하는 테이블을 저장합니다.

SYSAUX 테이블스페이스는 SYSTEM 테이블스페이스의 보조 테이블스페이스입니다. SYSAUX 테이블스페이스에는 많은 데이터베이스 구성 요소가 저장되어 있으므로 모든 데이터베이스 구성 요소가 올바르게 작동하려면 이 테이블스페이스가 온라인 상태여야 합니다. SYSTEM 및 SYSAUX 테이블스페이스는 응용 프로그램의 데이터를 저장하는 데 사용하지 않는 것이 좋습니다. 이 용도로 추가 테이블스페이스를 생성할 수 있습니다.

**주:** SYSAUX 테이블스페이스는 테이블스페이스 Recovery를 수행하기 위해 오프라인으로 전환할 수 있지만, SYSTEM 테이블스페이스의 경우에는 그렇게 할 수 없습니다. 그리고 두 테이블스페이스 중 어느 쪽도 읽기 전용으로 설정할 수 없습니다.

## 자동 저장 영역 관리(ASM)

- 이식 가능한 고성능 클러스터 파일 시스템
- 오라클 데이터베이스 파일을 관리합니다.
- ACFS(ASM 클러스터 파일 시스템)으로 응용 프로그램 파일을 관리합니다.
- 로드가 균형을 이루도록 여러 디스크에 데이터를 분산시킵니다.
- Failure 시 데이터를 Mirroring합니다.
- 저장 영역 관리 문제를 해결합니다.



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 자동 저장 영역 관리(ASM)

자동 저장 영역 관리(ASM)는 오라클 데이터베이스 파일을 위한 볼륨 관리자와 파일 시스템의 수직 통합을 제공합니다. ASM은 단일 SMP 시스템의 관리 기능이나 Oracle RAC(Real Application Clusters) 지원을 위해 클러스터의 여러 노드 간에 관리 기능을 제공합니다.

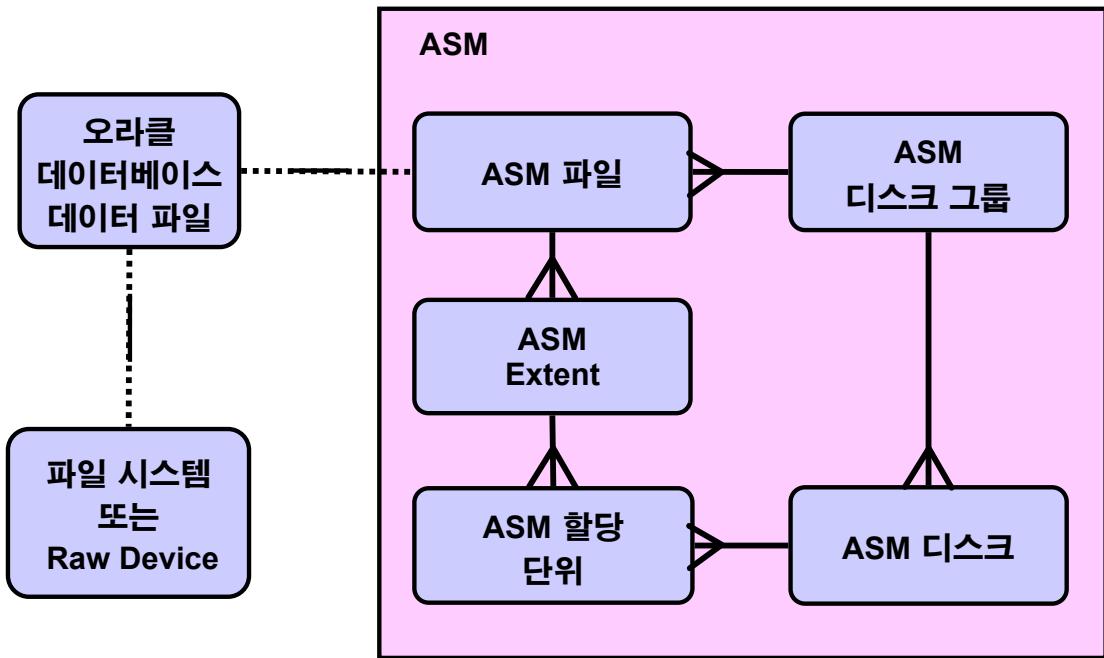
Oracle ACFS(ASM 클러스터 파일 시스템)는 다중 플랫폼, 확장 가능한 파일 시스템 및 저장 영역 관리 기술로서 실행 파일, 보고서, BFILE, 비디오, 오디오, 텍스트, 이미지 및 기타 범용 응용 프로그램 파일 데이터와 같은 오라클 데이터베이스 이외의 응용 프로그램 파일을 지원할 수 있도록 ASM 기능을 확장합니다.

ASM은 사용 가능한 모든 리소스에 I/O 로드를 분산하여 성능을 최적화하면서 수동 I/O 튜닝의 필요성도 제거합니다. ASM을 통해 DBA는 저장 영역 할당을 조정할 때 데이터베이스를 종료하지 않고 데이터베이스 크기를 늘리도록 허용함으로써 동적 데이터베이스 환경을 관리할 수 있습니다.

ASM은 결함 발생 시 대처할 수 있는 데이터의 중복 사본을 관리할 수 있으며 업체에서 공급한 신뢰할 수 있는 저장 영역 메커니즘의 최상위에 ASM을 구축할 수 있습니다. 데이터 관리는 파일별로 하나하나 유저가 상호 작용하는 것이 아니라 데이터 클래스에 대해 원하는 신뢰성 및 성능 특성을 선택하여 수행됩니다.

ASM 기능은 수동 저장 영역을 자동화함에 따라 더욱 많은 대규모 데이터베이스를 보다 향상된 효율성으로 관리할 수 있도록 함으로써 DBA의 시간을 절약합니다.

# ASM 저장 영역 구성 요소



ORACLE

Copyright © 2010, Oracle. All rights reserved.

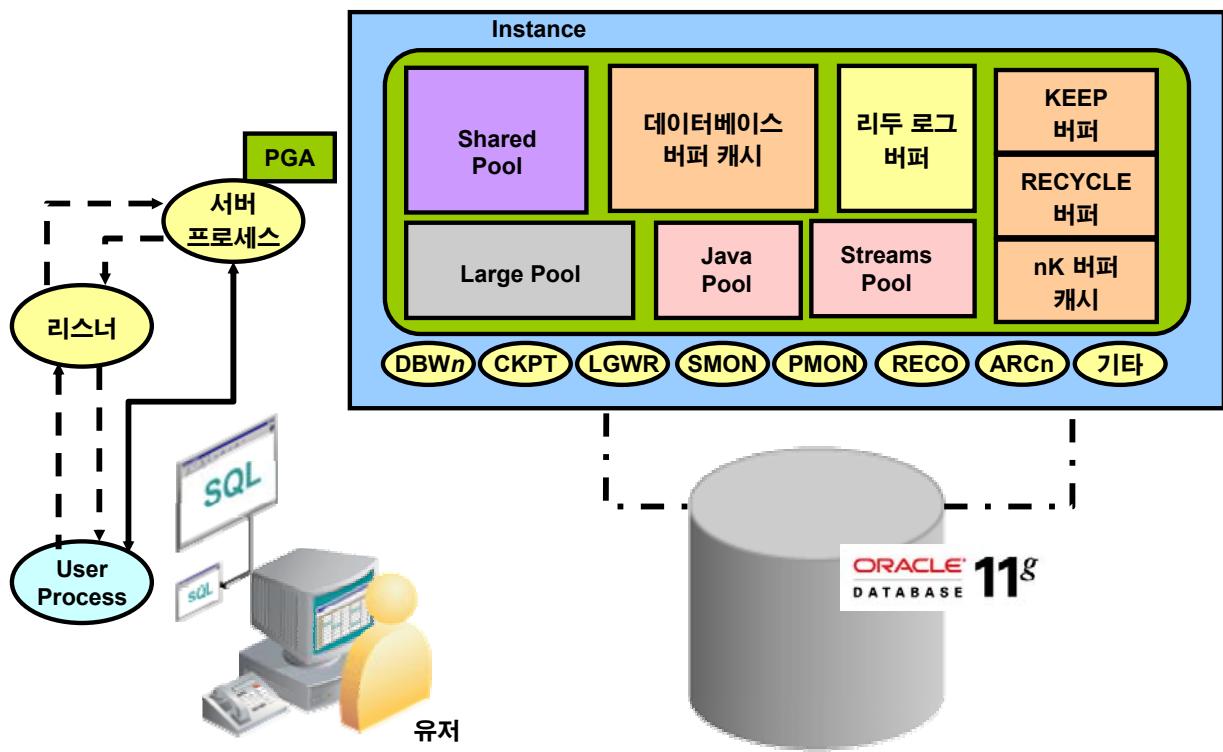
## ASM 저장 영역 구성 요소

ASM은 기존 데이터베이스 기능을 제거하지 않습니다. 기존의 데이터베이스는 평상시대로 작동할 수 있습니다. 새 파일은 ASM 파일로 생성하고 기존 파일은 이전 방식으로 관리하거나 ASM으로 이전할 수 있습니다.

도표에서는 오라클 데이터베이스 데이터 파일과 ASM 저장 영역 구성 요소 사이의 관계를 보여 줍니다. 새 파일 모양 표기는 일 대 다 관계를 나타냅니다. 오라클 데이터베이스 데이터 파일은 파일 시스템 또는 ASM 파일의 운영 체제에 저장된 파일과 일 대 일 관계를 갖습니다.

Oracle ASM 디스크 그룹은 논리적 단위로 관리되는 하나 이상의 Oracle ASM 디스크의 모음입니다. 디스크 그룹의 데이터 구조는 메타 데이터 요구를 위한 일부 공간을 사용하는 자체 포함 구조입니다. Oracle ASM 디스크는 Oracle ASM 디스크 그룹에 프로비전된 저장 장치이며, 물리적인 디스크 또는 Partition, 저장 영역 배열의 LUN(Logical Unit Number), LV(논리 블롭) 또는 네트워크 연결 파일일 수 있습니다. 각 ASM 디스크는 ASM이 할당하는 가장 작은 연속된 디스크 공간인 여러 ASM 할당 단위(AU: Allocation Unit)로 나뉩니다. ASM 디스크 그룹을 생성하면 디스크 그룹의 호환성 레벨에 따라 ASM 할당 단위 크기를 1, 2, 4, 8, 16, 32 또는 64MB로 설정할 수 있습니다. ASM Extent는 하나 이상의 ASM 할당 단위로 이뤄집니다. Oracle ASM Extent는 Oracle ASM 파일의 컨텐트를 보관하는 데 사용되는 Raw 저장 영역입니다. Oracle ASM 파일은 하나 이상의 파일 Extent로 구성됩니다. 1\*AU 크기, 4\*AU 크기 및 16\*AU 크기의 가변 Extent 크기는 초대형 ASM 파일을 지원하기 위해 사용됩니다.

# 오라클 데이터베이스와 상호 작용: 메모리, 프로세스 및 저장 영역



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스와 상호 작용

다음 예제에서는 가장 기본적인 레벨의 오라클 데이터베이스 작업을 설명합니다. 여기서는 유저와 관련 서버 프로세스가 별도의 컴퓨터에 있으며 네트워크를 통해 연결된 오라클 데이터베이스 구성은 보여줍니다.

1. 호스트 또는 데이터베이스 서버라고 하는 오라클 데이터베이스가 설치되어 있는 노드에서 Instance가 시작됩니다.
2. 유저가 User Process를 생성하는 응용 프로그램을 시작합니다. 응용 프로그램에서 서버에 대한 연결 설정을 시도합니다. 연결은 로컬, 클라이언트/서버 또는 Middle Tier로부터의 3계층 연결일 수 있습니다.
3. 서버는 해당하는 Oracle Net 서비스 처리기를 포함하는 리스너를 실행합니다. 리스너는 응용 프로그램의 연결 요청을 감지하고 User Process를 대신하여 Dedicated Server 프로세스를 생성합니다.
4. 유저가 DML 유형 SQL 문을 실행하고 트랜잭션을 커밋합니다. 예를 들면, 유저가 테이블에서 고객의 주소를 변경하고 변경 내용을 커밋합니다.
5. 서버 프로세스에서 해당 명령문을 받아 Shared Pool(SGA 구성 요소)에서 동일한 SQL 문이 포함된 공유 SQL 영역이 있는지 검사합니다. 공유 SQL 영역이 발견되면 서버 프로세스는 요청된 데이터에 대한 유저의 액세스 권한을 확인하고 기존 공유 SQL 영역이 명령문 처리에 사용됩니다. 공유 SQL 영역이 발견되지 않으면 명령문이 구문 분석 및 처리되도록 새 공유 SQL 영역이 명령문에 할당됩니다.

## 오라클 데이터베이스와 상호 작용(계속)

6. 서버 프로세스는 실제 데이터 파일(테이블) 또는 데이터베이스 버퍼 캐시에 저장된 값에서 필요한 데이터 값을 검색합니다.
7. 서버 프로세스가 SGA의 데이터를 수정합니다. 트랜잭션이 커밋되었으므로 LGWR(로그 기록자) 프로세스가 즉시 트랜잭션을 리두 로그 파일에 기록합니다. DBW $n$ (데이터베이스 기록자) 프로세스는 가능한 경우 수정된 블록을 디스크에 영구적으로 기록합니다.
8. 트랜잭션이 성공하면 서버 프로세스가 네트워크를 통해 메시지를 응용 프로그램으로 보냅니다. 그리고 트랜잭션이 실패하면 오류 메시지가 전송됩니다.
9. 이러한 전체 과정에서 개입이 필요한 상황이 되면 다른 백그라운드 프로세스가 실행됩니다. 또한 데이터베이스 서버는 다른 유저의 트랜잭션을 관리하고 동일한 데이터를 요청하는 트랜잭션 간의 경합을 방지합니다.

## 퀴즈

**PMON(프로세스 모니터 프로세스):**

- 1. Instance 시작 시 Recovery 수행**
- 2. User Process가 실패할 경우 프로세스 Recovery 수행**
- 3. 모든 In-Doubt 트랜잭션을 자동으로 해결**
- 4. 리두 로그 버퍼를 리두 로그 파일에 기록합니다.**

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

**해답: 2**

## 퀴즈

ASM 파일은 어떤 유형의 Instance에서 액세스됩니까?

1. RDBMS Instance만
2. ASM Instance만
3. RDBMS 및 ASM Instance 모두

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 3

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 오라클 데이터베이스의 주요 구조적 구성 요소 나열
- 메모리 구조 설명
- 백그라운드 프로세스 설명
- 논리적/물리적 저장 영역 구조 상호 연관
- ASM 저장 영역 구성 요소 설명

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 1: 개요

이 연습은 다음에 대한 질문이 포함된 서면 연습입니다.

- 데이터베이스 구조
- 메모리
- 프로세스
- 파일 구조



Copyright © 2010, Oracle. All rights reserved.



# Oracle 소프트웨어 설치

Copyright © 2010, Oracle. All rights reserved.

ORACLE®

## 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- DBA(데이터베이스 관리자)로서의 역할과 일반적인 작업 및 도구 설명
- 오라클 데이터베이스 설치 계획
- 독립형 서버를 위한 Oracle 그리드 Infrastructure 설치
- 오라클 데이터베이스 소프트웨어 설치

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 오라클 데이터베이스 관리자의 작업

오라클 데이터베이스의 설계, 구현 및 유지 관리에 대한 접근 방식은 다음 작업과 관련되어 있습니다.

1. 데이터베이스 서버 하드웨어 평가
2. Oracle 소프트웨어 설치
3. 데이터베이스 및 보안 전략 계획
4. 데이터베이스 생성, 이전 및 열기
5. 데이터베이스 백업
6. 시스템 유저 등록 및 해당 유저의 Oracle 네트워크 액세스 계획
7. 데이터베이스 설계 구현
8. 데이터베이스 failure로부터 recovery
9. 데이터베이스 성능 모니터



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스 관리자의 작업

Oracle 소프트웨어 설치 및 데이터베이스 생성 업무는 일반적으로 DBA가 담당합니다.

DBA는 테이블스페이스와 같은 데이터베이스 저장 영역 구조를 생성하는 업무를 담당할 수도 있습니다. 또한 응용 프로그램 데이터를 보유하는 스키마나 객체 집합을 생성할 수도 있습니다.

DBA는 데이터베이스를 시작하고, 정기적으로 백업하고, 성능을 모니터하여 유저가 데이터베이스를 사용할 수 있도록 준비해야 합니다. 이러한 작업은 보안 전략의 프레임워크 내에서 수행되어야 합니다.

이 과정의 각 단원에서는 이러한 작업을 수행하는 방법에 대해 설명합니다. 이 슬라이드에 요약된 각 작업에 대한 자세한 정보는 *Oracle Database Administrator's Guide*를 참조하십시오.

이 단원에서는 설치 과정을 중점적으로 다룹니다. 설치 작업과 관련하여 다음과 같은 세부 고려합니다. 작업을

- 조직의 전반적인 기술 구조에 적합하도록 설치하는 방법 이해
- 용량 계획 검토 및 개선
- 데이터베이스 소프트웨어(필요한 버전 및 옵션) 선택
- 선택한 모든 요소에 대해 시스템 요구 사항이 충족되는지 확인

# 오라클 데이터베이스 관리 도구

- **Oracle Universal Installer**
- **Database Configuration Assistant**
- **Database Upgrade Assistant**
- **Oracle Net Manager**
- **Oracle Net Configuration Assistant**
- **Oracle Enterprise Manager**
- **Server Control 유틸리티**
- **SQL\*Plus**
- **Recovery Manager**
- **Data Pump**
- **SQL\*Loader**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스 관리 도구

다음과 같은 설치 및 업그레이드용 도구를 사용할 수 있습니다.

- **OUI(Oracle Universal Installer):** Oracle 소프트웨어 및 옵션을 설치하며, Database Configuration Assistant를 자동으로 실행하여 데이터베이스를 생성할 수 있습니다.
- **DBCA(Database Configuration Assistant):** 오라클 제공 템플리트에서 데이터베이스를 생성하여 미리 구성된 시드 데이터베이스를 복사할 수 있도록 합니다. 또는 데이터베이스와 템플리트를 직접 생성할 수도 있습니다.
- **DBUA(Database Upgrade Assistant):** 기존 데이터베이스를 새 Oracle 릴리스로 업그레이드하는 과정을 안내합니다.
- **Oracle Net Manager(`netmgr`):** 오라클 데이터베이스 및 응용 프로그램에 대해 네트워크 연결 기능을 구성합니다.
- **NetCA(Oracle Net Configuration Assistant):** Oracle 네트워크 구성을 구성 및 관리하는데 사용되는 마법사 기반의 그래픽 도구입니다.

## 오라클 데이터베이스 관리 도구 (계속)

다음은 Oracle instance 및 데이터베이스 관리에 사용되는 도구입니다.

- **Oracle EM(Enterprise Manager):** 그래픽 콘솔, 에이전트, 공통 서비스 및 도구를 결합하여 Oracle 제품을 관리하는 포괄적인 통합 시스템 관리 플랫폼을 제공합니다. Oracle 소프트웨어를 설치하고 데이터베이스를 생성하거나 업그레이드하고 네트워크를 구성한 후에, EM을 데이터베이스 관리를 위한 단일 인터페이스로 사용할 수 있습니다. EM은 SQL 명령을 실행하기 위한 웹 기반 유저 인터페이스를 제공하는 것은 물론 데이터베이스를 관리하는 데 사용되는 다른 Oracle 구성 요소(예: Recovery Manager, Scheduler)와도 완벽하게 통합됩니다.
- 오라클 데이터베이스를 관리하는 데 사용되는 기본 EM 도구는 다음과 같습니다.
  - **Enterprise Manager Database Console:** 단일 데이터베이스를 관리하는 데 사용됩니다.
  - **Enterprise Manager Grid Control:** 다수의 데이터베이스를 동시에 관리하는 데 사용됩니다.
- **Server Control 유틸리티(srvctl):** 데이터베이스 및 Instance를 시작 및 정지하고, ASM Instance를 관리하고, 구성 정보를 관리하고, Instance 및 서비스를 이동 또는 제거하는 데 사용할 수 있는 표준 명령행 인터페이스입니다. SRVCTL을 사용하여 서비스를 추가하고 구성 정보를 관리할 수도 있습니다.
- **SQL\*Plus:** 데이터베이스 관리에 사용되는 표준 명령행 인터페이스입니다.
- **RMAN(Recovery Manager):** 전체 데이터베이스 또는 특정 데이터베이스 파일의 백업, 복원 및 Recovery에 필요한 모든 솔루션을 제공하는 Oracle 도구입니다.
- **Data Pump:** 데이터베이스 간에 데이터를 빠른 속도로 전송할 수 있게 해줍니다. 예를 들어, 테이블을 엑스포트하여 다른 데이터베이스로 임포트해야 하는 경우에 활용할 수 있습니다.
- **SQL\*Loader:** 데이터를 외부 파일에서 오라클 데이터베이스로 로드하는 데 사용할 수 있으며, 데이터베이스 테이블로 데이터를 로드하는 데 사용할 수 있는 여러 오라클 유틸리티 중 하나입니다.
- **명령행 도구:**
  - Enterprise Manager를 관리하려면 다음 명령을 사용합니다.  
`emctl start | status | stop dbconsole`
  - 리스너를 관리하려면 다음 명령을 사용합니다.  
`lsnrctl start | status | stop`

## 설치 계획

- 어떤 Oracle 소프트웨어를 설치할 것인가?
- 관련된 하드웨어가 최소 필요 사양을 충족하는가?
- 여러 제품이 관련된 경우 권장되는 설치 순서가 있는가?
- DBA 이외의 다른 유저가 수행해야 하는 선행 단계가 있는가?



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 설치 계획

Oracle 소프트웨어 설치를 시작하기 전에 설치를 계획하는 데 도움이 될 수 있는 다음과 같은 항목들을 먼저 고려해야 합니다.

- 어떤 Oracle 소프트웨어를 설치할 것인가?

본 과정에서는 오라클 데이터베이스를 설치한다고 가정합니다. 오라클은 저장 영역 기술로 자동 저장 영역 관리(ASM)를 사용할 것을 권장합니다. 이를 위해서는 클러스터 소프트웨어 Media에서 Oracle 그리드 Infrastructure를 설치해야 합니다. 이를 설치하면 Oracle Restart는 물론 ASM에 필요한 구성 요소가 설치됩니다.

- 관련된 하드웨어가 최소 필요 사양을 충족하는가?

설치 프로세스와 관련되는 모든 하드웨어를 식별하고 최소 권장 사양이 충족되는지 확인합니다.

- 여러 제품이 관련된 경우 권장되는 설치 순서가 있는가?

가능하다면 언제나 Oracle 그리드 Infrastructure를 오라클 데이터베이스 소프트웨어보다 먼저 설치하는 것이 좋습니다. 이 순서대로 설치하면 새로 생성되는 데이터베이스가 ASM 디스크 그룹을 사용할 수 있도록 구성되고 이 데이터베이스가 Oracle Restart에 자동으로 등록됩니다. Oracle 그리드 Infrastructure를 오라클 데이터베이스 다음에 설치할 경우에는 해당 데이터베이스를 Oracle Restart에 등록하기 위해 수동 구성 단계가 필요합니다. 이 기존 데이터베이스가 ASM 디스크 그룹을 저장 영역으로 사용하도록 하려면 이전 단계가 필요합니다.

## 설치 계획 (계속)

- **DBA 이외의 다른 유저가 수행해야 하는 실행 단계가 있는가?**

권장되는 책임 구분 지침을 따를 경우 DBA는 설치에 사용할 하드웨어 및 저장 장치를 구성할 책임이 없을 것입니다. Oracle 그리드 Infrastructure를 설치하기 전, 필요한 디스크 Partition을 구성하기 위한 필수 구성 단계는 저장 영역 관리자가 수행해야 할 것입니다. 자세한 사항은 Oracle Database Installation Guide를 참조하십시오.

# Oracle 그리드 Infrastructure 및 오라클 데이터베이스 설치: 시스템 요구 사항

- 메모리 요구 사항:
  - Oracle Enterprise Manager Database Control을 포함하는 데이터베이스 Instance의 경우 1GB
  - ASM Instance 및 Oracle Restart의 경우 1.5GB
- 디스크 공간 요구 사항:
  - 3GB의 스왑 스페이스(2GB RAM 기준)
  - 1GB의 /tmp 디렉토리 디스크 공간
  - 3.8GB의 오라클 데이터베이스 소프트웨어용 공간
  - 4.5GB의 그리드 Infrastructure 소프트웨어용 공간
  - 1.7GB의 미리 구성된 데이터베이스용 공간(선택 사항)
  - 3.4GB의 Fast Recovery Area(선택 사항)
- 운영 체제(설명서 참조)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Oracle 그리드 Infrastructure 및 오라클 데이터베이스 설치: 시스템 요구 사항

- 표준 데이터베이스 설치를 완료하려면 컴퓨터에 1GB의 RAM과 1.5GB 이상의 스왑 스페이스가 있어야 합니다. 독립형 서버 설치를 위한 표준 Oracle 그리드 Infrastructure에는 최소 1.5GB RAM이 필요합니다.
- 필요한 스왑 스페이스 양은 RAM 용량에 따라 달라집니다(예: 2GB RAM의 경우 3GB의 스왑 스페이스 필요). 자세한 내용은 OS별 설치 설명서를 참조하십시오.
- 오라클 데이터베이스 소프트웨어를 설치하는 시스템의 사용 정도에 따라 20분 이내에 표준 설치를 완료할 수 있습니다.
- 설치 세부 사항:
  - Oracle Database 11g에는 두 개의 시드 데이터베이스 템플릿이 포함됩니다.
  - 중복된 파일은 제거됩니다.
  - 추가 CD를 통해 다수의 기타 제품과 데모를 설치할 수 있습니다.

슬라이드에 나열된 하드웨어 요구 사항은 모든 플랫폼에서 최소 요구 사항입니다.

설치 시 디스크 공간 등과 같은 추가 요구 사항이 있을 수도 있습니다.

주: 표준 시드 데이터베이스가 포함된 Enterprise Edition 설치 유형을 "표준 설치"라고 합니다.

# 운영 체제 준비

## 필요한 운영 체제 유저 및 그룹 생성:

- **그룹:**
  - oinstall
  - dba
  - 선택적인 그룹(여러 유저 간의 책임 구분을 사용하는 경우):
    - oper
    - asmdba
    - asmoper
    - asmadmin
- **유저:**
  - 소프트웨어 소유자, 일반적으로 oracle
  - 여러 제품 설치에 대해 여러 유저를 생성할 수 있음

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 운영 체제 준비

오라클 설치를 위해 사용하려는 하드웨어에 대해서는 시스템 관리자가 수행해야 하는 단계가 있습니다. 모든 운영 체제 구성 세부 정보 및 명령을 다루는 것은 이 과정의 범위를 벗어납니다. 운영 체제마다 Oracle 소프트웨어를 설치하기 위한 요구 사항도 다릅니다. 대신 이 과정에서는 몇 가지 간략한 요구 사항에 대해서만 설명하고, Oracle 설치를 위해서는 해당 운영 체제별 설치 설명서를 참조할 것을 권장합니다.

필요한 단계 중 하나는 필요한 운영 체제 그룹 및 유저를 생성하는 것입니다. 필수 운영 체제 그룹은 oinstall 및 dba입니다. 여러 유저 계정 간에 책임 구분을 구현할 경우 oper, asmdba, asmoper 및 asmadmin도 생성해야 합니다. 적어도 하나의 운영 체제 유저는 오라클 설치의 소유자가 되어야 합니다. 대부분의 경우에는 oracle 유저가 이 용도로 구성됩니다. 책임 구분을 정확하게 나누려면 오라클 제품별로 다른 소유자를 설정하면 됩니다.

## 환경 변수 설정

### 오라클 환경 변수:

- **ORACLE\_BASE:** 오라클 디렉토리 구조의 기반입니다.  
이 변수는 설치 전에 설정하는 것이 좋습니다.
- **ORACLE\_HOME:** 오라클 제품이 실행되는 환경입니다.  
ORACLE\_BASE가 설정된 경우 설치 전에 설정할 필요가 없습니다.
- **ORACLE\_SID:** 설치 전에 필수는 아니지만 특정 Instance 와의 상호 작용을 쉽게 할 수 있으므로 유용합니다.
- **NLS\_LANG:** 언어, 지역 및 클라이언트 Character Set 설정을 제어하는 선택적인 환경 변수입니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 환경 변수 설정

각 오라클 환경에는 여러 오라클 환경 변수가 관련되어 있습니다. 여기서 언급하는 환경 변수는 오라클 데이터베이스를 올바르게 설치하고 사용하는 데 중요한 역할을 합니다. 이러한 변수를 반드시 설정해야 하는 것은 아니지만 설정하면 앞으로 발생할 수 있는 문제를 피할 수 있습니다.

- **ORACLE\_BASE:** Oracle 지원 센터에서 권장하고 있는 OFA(Optimal Flexible Architecture)에 대한 Oracle 디렉토리 구조의 기반을 지정합니다. 이 변수의 사용은 선택 사항이지만, 사용할 경우 이후의 설치 및 업그레이드를 쉽게 수행할 수 있습니다. 이 변수는 다음 예제에서 보여주듯이 디렉토리 경로입니다.  
`/u01/app/oracle`
- **ORACLE\_HOME:** 오라클 제품이 실행되는 환경입니다. ORACLE\_BASE가 설정된 경우 설치 전에 설정할 필요가 없습니다. OUI는 ORACLE\_BASE 설정을 사용하여 유저의 설치에 권장되는 ORACLE\_HOME을 결정할 수 있습니다. 이 환경 변수를 설정하면 Oracle 소프트웨어를 보다 쉽게 유지 관리 및 관리할 수 있습니다. 이 변수는 다음 예제에서 보여주듯이 디렉토리 경로입니다.  
`/u01/app/oracle/product/11.2.0/dbhome_1`
- **ORACLE\_SID:** Oracle Instance에 대한 시스템 식별자입니다(예: 데이터베이스의 경우 orcl, ASM Instance의 경우 +ASM). 설치 전에 필수는 아니지만 특정 Instance와의 상호 작용을 쉽게 할 수 있으므로 유용합니다.
- **NLS\_LANG:** 다음 예제에서 보여주듯이 언어, 지역 및 클라이언트 Character Set 설정을 제어하는 선택적인 환경 변수입니다.

`AMERICAN_DENMARK.WE8MSWIN1252`  
올바른 언어, 지역(territory), Character Set 및 언어 지원에 대한 자세한 사항은 Oracle Database Globalization Support Guide를 참조하십시오.

# 시스템 요구 사항 검사

- 적절한 임시 영역
- 64비트와  
32비트 문제
- 올바른 OS(운영 체제)
- OS 패치 레벨
- 시스템 패키지
- 시스템 및 커널 파라미터
- X 서버 사용 권한
- 충분한 스와핑
- ORACLE\_HOME 상태

```
[oracle@edrsrl2p1-+ASM Disk1]$ ./runInstaller
Starting Oracle Universal Installer...
Checking Temp space: must be greater than 80 MB.  Actual 15067 MB  Passed
Checking swap space: must be greater than 150 MB.  Actual 4000 MB  Passed
Checking monitor: must be configured to display at least 256 colors.  Actual 6
5536  Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2009-05-15_12
-04-10AM. Please wait ...
```

ORACLE®

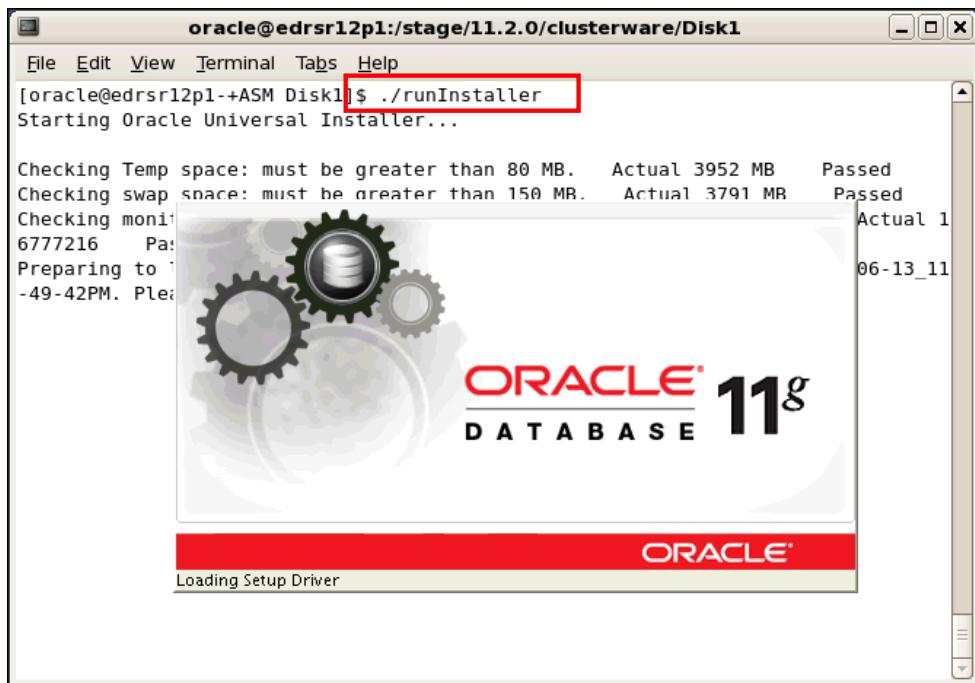
Copyright © 2010, Oracle. All rights reserved.

## 시스템 요구 사항 검사

Oracle Universal Installer는 대부분의 필수 구성 요소 검사를 자동으로 수행하여 다음을 확인합니다.

- 설치 및 구성을 위한 최소 임시 공간 요구 사항을 검사합니다. 이러한 요구 사항은 설치 프로세스가 진행되는 중에 확인됩니다.
- Oracle 홈에 이미 32비트 소프트웨어가 설치된 경우 64비트 설치가 허용되지 않습니다. 그 반대의 경우도 마찬가지입니다.
- Oracle Grid Infrastructure 11g 및 Oracle Database 11g는 Linux 플랫폼의 여러 버전 및 기타 플랫폼에서 작동이 검증되었습니다.
- 필수 OS 패치가 모두 설치되어 있습니다.
- 모든 필수 시스템 및 커널 파라미터가 올바르게 설정되어 있습니다.
- DISPLAY 환경 변수가 설정되어 있고 유저가 지정된 DISPLAY를 표시하기 위한 충분한 권한을 가지고 있습니다.
- 시스템에 스와핑 세트가 충분합니다.
- 새로운 설치를 위한 Oracle 홈이 비어 있거나, Oracle Database 11g를 설치할 수 있는 일부 지원되는 릴리스 중 하나입니다. 또한 설치 프로세스에서는 이러한 릴리스가 Oracle Inventory에 등록되어 있는지 확인합니다.

# OUI(Oracle Universal Installer)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## OUI(Oracle Universal Installer)

OUI(Oracle Universal Installer)는 구성 요소 기반 설치를 수행하고 단일 패키지의 복잡한 논리뿐만 아니라 다양한 레벨의 통합 번들 및 제품군을 설치하고 웹 기반 설치를 수행하는 Java 응용 프로그램입니다. 모든 Java 사용 플랫폼에 이 설치 엔진을 쉽게 이식할 수 있으며 전체 설치 프로세스에서 플랫폼별 문제를 캡슐화할 수 있습니다.

OUI는 소프트웨어 관리 및 배포 문제를 해결하기 위해 다음과 같은 기능을 제공합니다.

- 자동 종속성 확인 및 복잡한 논리 처리
- 웹을 통한 설치
- 구성 요소 및 제품군 설치
- 암시적 제거
- 여러 Oracle 홈 지원
- NLS 또는 Globalization 지원
- 분산된 설치 지원
- 응답 파일을 사용한 무인 "자동" 설치

## 예제: 설치 시나리오

이 단원에서 제공되는 설치 시나리오는 다음 두 부분으로 나뉩니다.

- 1부: 독립형 서버를 위한 Oracle 그리드 Infrastructure 설치
- 2부: 오라클 데이터베이스 소프트웨어 설치

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

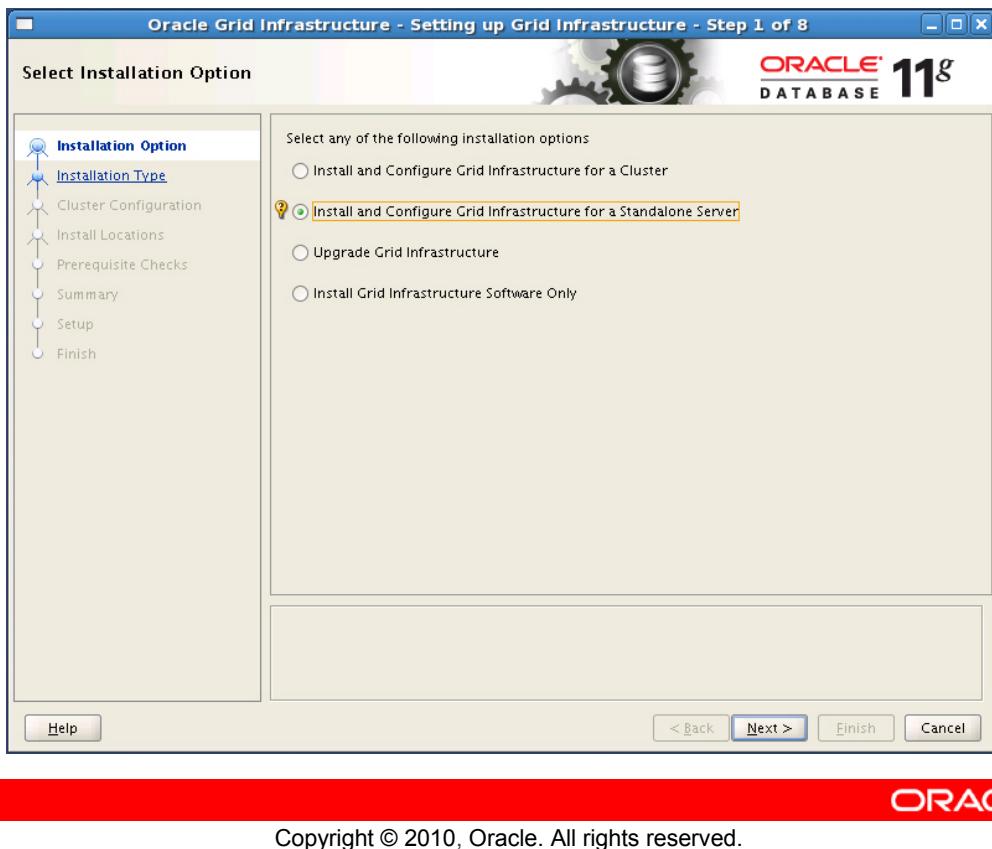
### 예제: 설치 시나리오

이 단원에서 제공되는 설치 시나리오는 다음 두 부분으로 나뉩니다.

- 1부: 독립형 서버를 위한 Oracle 그리드 Infrastructure 설치
- 2부: 오라클 데이터베이스 소프트웨어 설치

Oracle 그리드 Infrastructure 설치 중에는 ASM 디스크 그룹을 구성하는 단계가 제시되고 Oracle Restart가 구성됩니다. Oracle 그리드 Infrastructure는 오라클 데이터베이스 소프트웨어 설치 후 생성되는 데이터베이스가 ASM 디스크 그룹을 사용할 수 있고 Oracle Restart에 자동으로 등록되도록 먼저 설치됩니다.

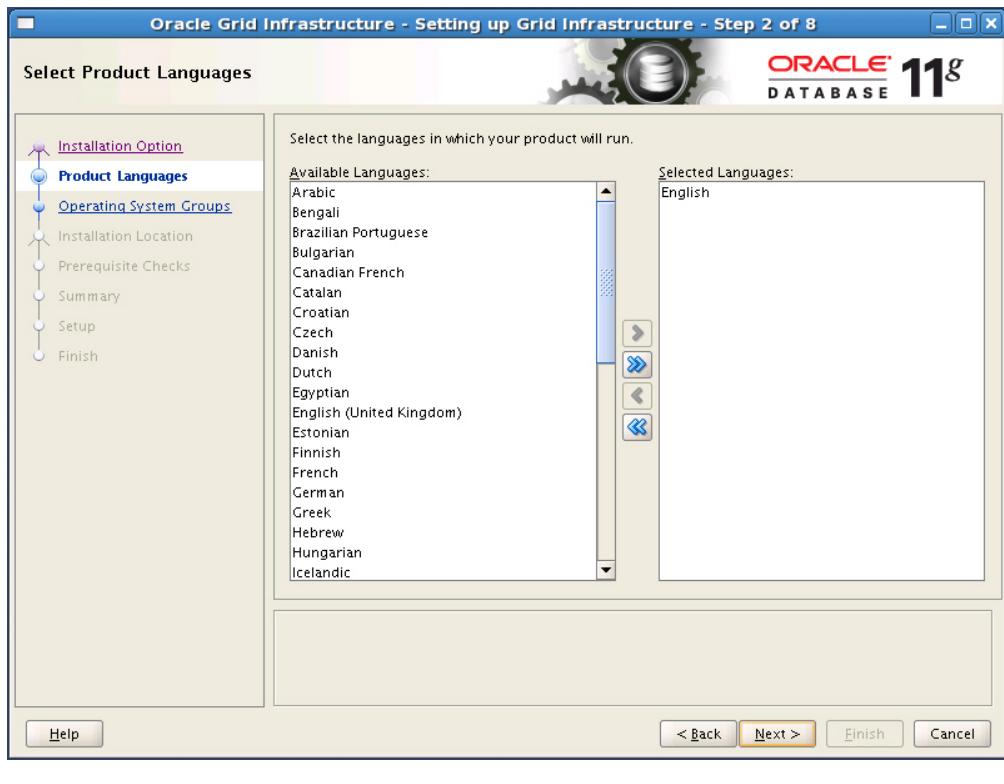
# 1부: 독립형 서버를 위한 Oracle Grid Infrastructure 설치



## 1부: 독립형 서버를 위한 Oracle Grid Infrastructure 설치

OUI(Oracle Universal Installer)를 사용하여 Oracle Grid Infrastructure 소프트웨어를 설치하려면 오라클 소프트웨어 설치 및 데이터베이스 생성과 관리가 승인된 관리 그룹의 멤버로 컴퓨터에 로그온합니다. 클러스터웨어 배포 CD를 CD 드라이브에 넣거나 Oracle Clusterware 스테이지 위치로 이동합니다. 이 위치에서 `./runInstaller`를 입력하여 OUI를 시작합니다. OUI의 Installation Option이 표시됩니다. Install and Configure Grid Infrastructure for a Standalone Server 옵션을 선택하고 Next를 누릅니다.

## 제품 언어 선택



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

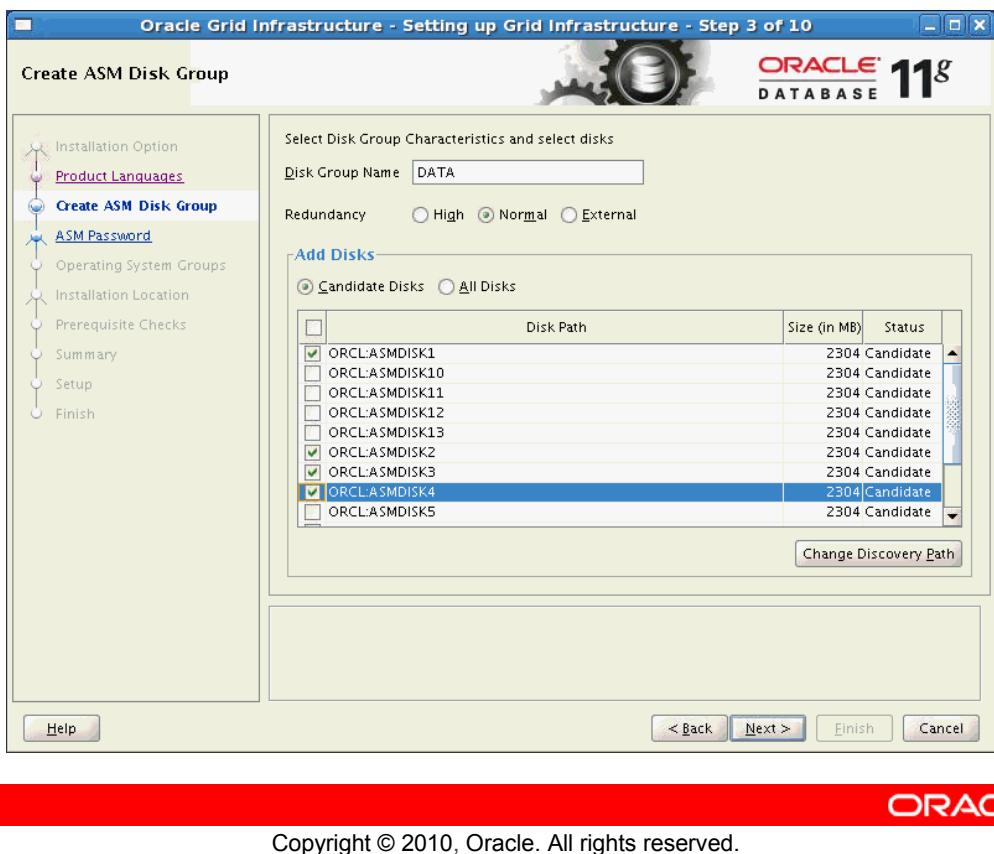
### 제품 언어 선택

"Select Product Languages" 페이지가 표시됩니다. 설치에 언어를 추가하려면 해당 언어를 눌러 강조 표시한 후 오른쪽 화살표 버튼을 사용하여 "Selected Languages" list로 이동합니다. Control 키를 누른 상태에서 마우스로 항목을 선택하면 여러 언어를 선택할 수 있습니다. Next 버튼을 눌러 설치를 계속합니다.

현재 언어 리스트에는 아랍어, 벵골어, 포르투갈어(브라질), 불가리아어, 프랑스어(캐나다), 카탈로니아어, 크로아티아어, 체코어, 덴마크어, 네덜란드어, 이집트어, 영어(영구), 에스토니아어, 핀란드어, 프랑스어, 독일어, 그리스어, 히브리어, 헝가리어, 아이슬란드어, 인도네시아어, 이탈리아어, 일본어, 한국어, 스페인어(라틴 아메리카), 라트비아어, 리투아니아어, 말레이어, 스페인어(멕시코), 노르웨이어, 폴란드어, 포르투갈어, 루마니아어, 러시아어, 중국어(간체), 슬로바키아어, 슬로베니아어, 스페인어, 스웨덴어, 태국어, 중국어(번체), 터키어, 우크라이나어 및 베트남어가 포함됩니다.

**주:** 이 리스트는 최신 내용으로 변경될 수 있습니다.

# ASM 디스크 그룹 생성



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM 디스크 그룹 생성

"Create ASM Disk Group" 페이지가 표시됩니다. Oracle Grid Infrastructure에는 ASM 및 Oracle Restart에 대한 지원이 포함됩니다. OUI는 ASM 디스크 그룹이 생성될 때까지 진행되지 않습니다. 생성할 첫번째 ASM 디스크 그룹의 이름을 "Disk Group Name" 필드에 입력합니다. "Add Disks" 섹션에 후보 디스크가 채워져 있지 않으면 "Change Discovery Path" 버튼을 누르고 ASM 디스크에 디스크 경로를 입력합니다.

오라클은 대부분의 환경에서 두 개의 디스크 그룹을 생성할 것을 권장합니다. OUI는 현재 단일 ASM 디스크 그룹만 생성할 수 있습니다. ASM Configuration Assistant(asmca) 툴리티 또는 SQL\*Plus를 사용하여 설치를 마친 다음에는 두번쩨 권장 ASM 디스크 그룹을 생성할 수 있습니다. Next 버튼을 눌러 설치를 계속합니다.

**주:** 이 과정의 예에서는 ASMLib를 구성 중이기 때문에 장치가 리스트에 표시됩니다. ASMLib는 Linux 플랫폼 전용입니다. 다른 플랫폼에서는 Change Discovery Path 버튼을 눌러야 할 수 있습니다.

# ASM 암호 정의



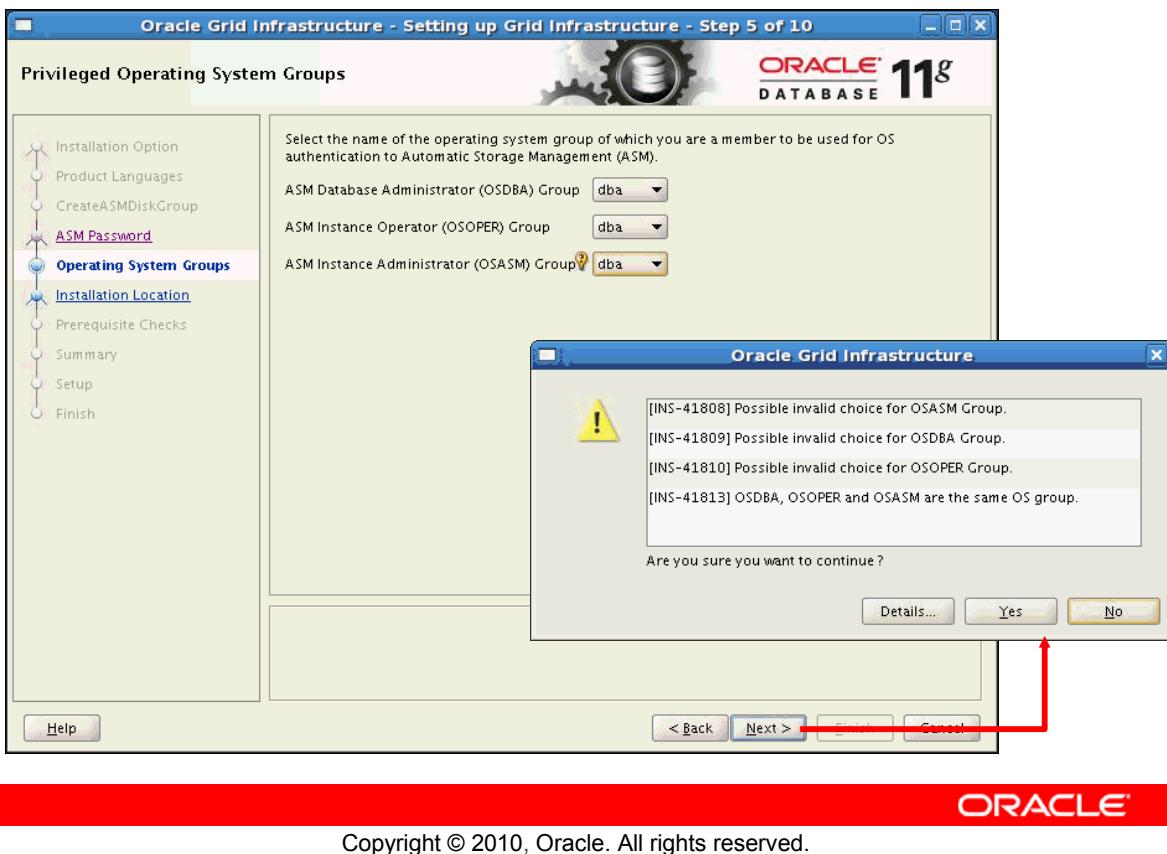
ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM 암호 정의

"Specify ASM Password" 페이지가 표시됩니다. SYS 계정 및 ASMSNMP 계정의 두 계정에 대해 암호를 제공해야 합니다. 계정에 대해 여러 암호를 사용하는 옵션과 계정에 대해 동일한 암호를 사용하는 옵션이 제공됩니다. ASM Instance는 오라클 데이터베이스와 같은 데이터 딕셔너리를 포함하지 않으므로 유일한 인증 방법은 운영 체제 인증과 Password file 인증입니다. SYS 계정은 Password file(Linux의 경우 orapw+ASM)에 추가되고 SYSDBA, SYSOPER 및 SYSASM 권한이 부여됩니다. ASMSNMP 계정은 Password file에 추가되고 SYSDBA 권한만 부여됩니다. 적합한 암호를 입력한 후에는 Next 버튼을 눌러 설치를 계속합니다.

## 권한이 있는 운영 체제 그룹 정의



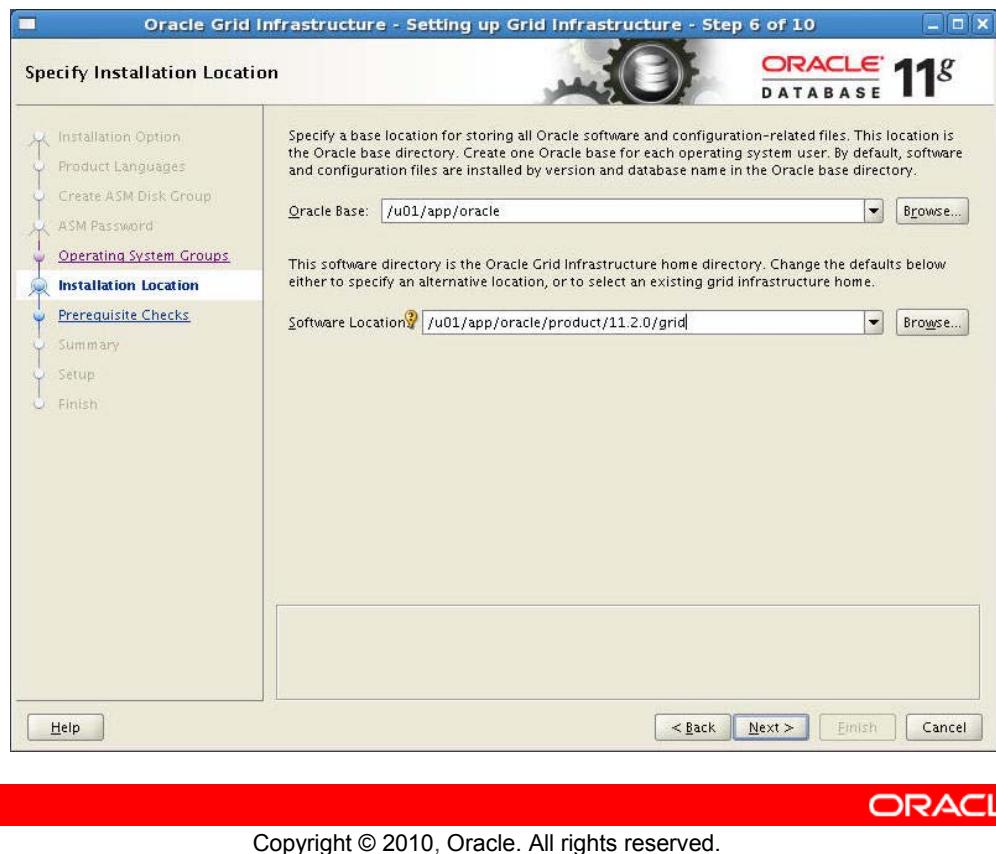
### 권한이 있는 운영 체제 그룹 정의

"Privileged Operating System Groups" 페이지가 표시됩니다. OUI 유트리티는 현재 유저가 그룹의 운영 체제 멤버인 경우 다음 그룹에 대한 권장 기본값을 제공합니다.

- ASM Database Administrator(OSDBA) 그룹 – asmdba
- ASM Instance Operator(OSOPER) 그룹 – asmoper
- ASM Instance Administrator(OSASM) 그룹 – asmadmin

이 설치는 독립형 서버용으로 이 세 개의 그룹 모두에 대해 dba(슬라이드에 표시된 대로)와 같은 동일 운영 체제 그룹을 사용하는 것이 일반적입니다. Next 버튼을 눌러 설치를 계속합니다. 이 과정에서는 OSDBA, OSOPER 및 OSASM에 대해 동일한 운영 체제 그룹을 사용하므로 경고가 표시되면 Yes를 누릅니다.

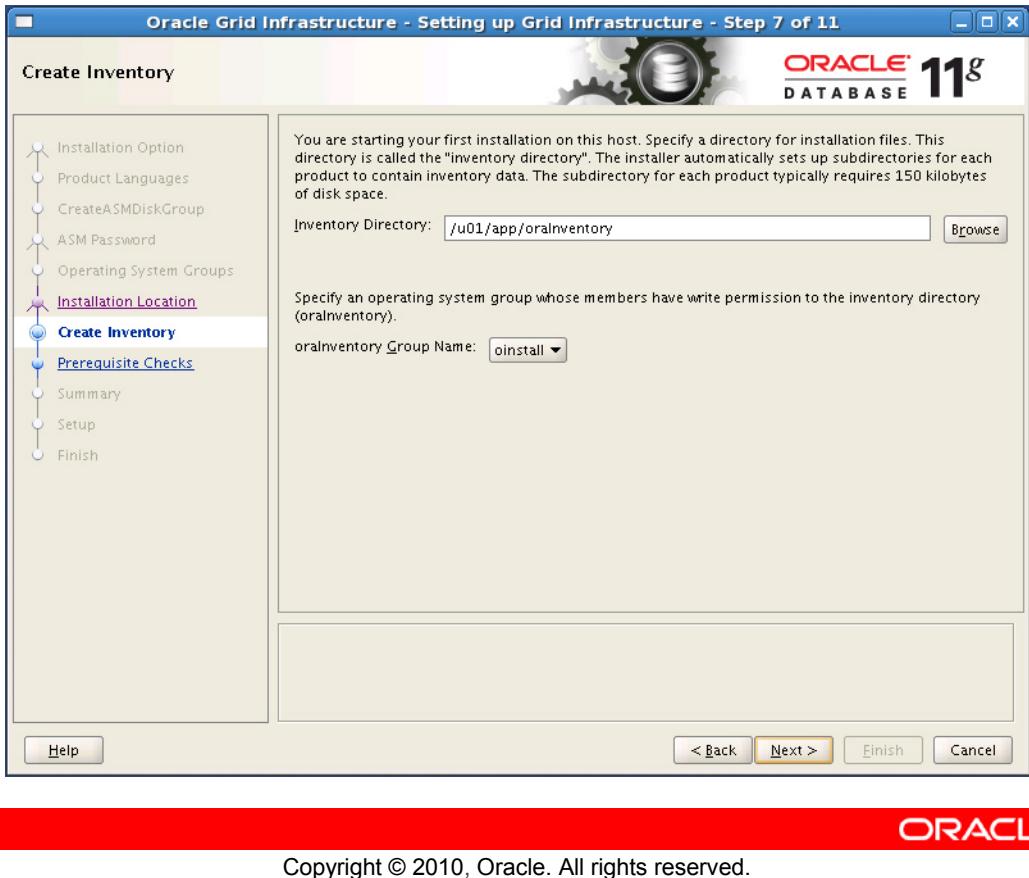
## 설치 위치 지정



### 설치 위치 지정

"Specify Installation Location" 페이지가 표시됩니다. "Oracle Base" 필드에는 소프트웨어 소유자의 ORACLE\_BASE 값을 입력합니다. 기본값은 /u01/app/oracle입니다. "Software Location" 필드에는 그리드 Infrastructure 소프트웨어의 ORACLE\_HOME 값을 입력합니다. 기본값은 /u01/app/oracle/product/11.2.0/grid입니다. Next 버튼을 눌러 설치를 계속합니다.

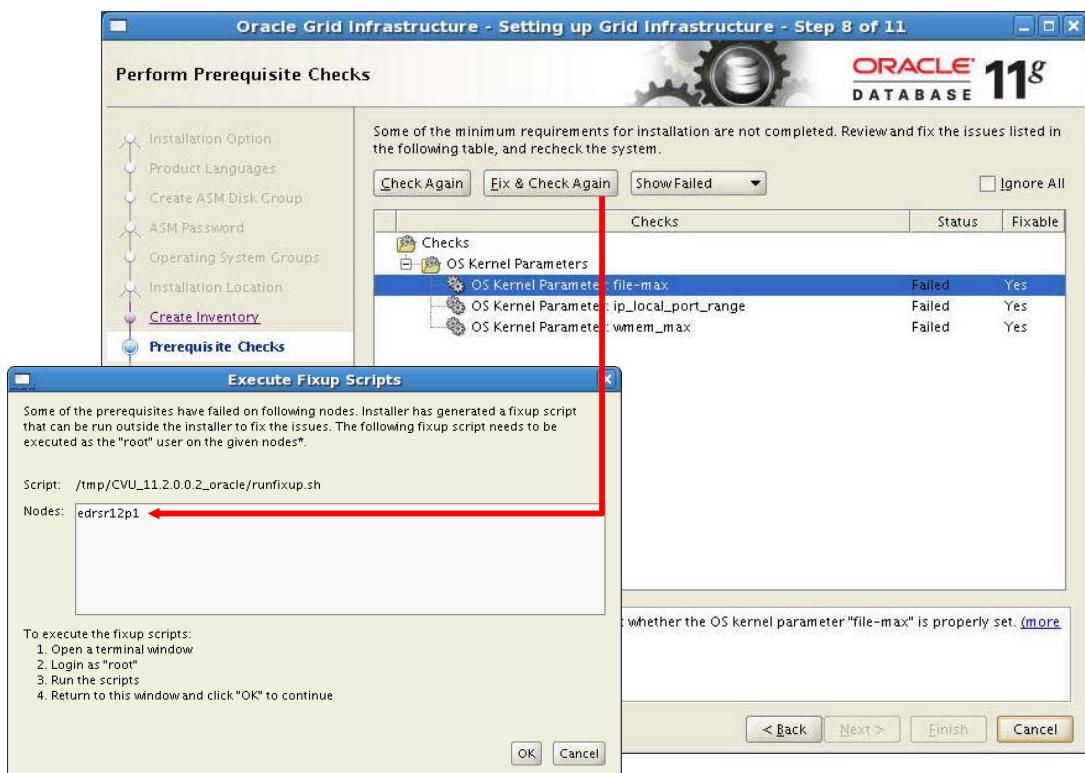
# Inventory 생성



## Inventory 생성

Oracle Central Inventory 디렉토리가 호스트 시스템에 존재하지 않으면 "Create Inventory" 페이지가 표시됩니다. "Inventory Directory" 필드에 oraInventory 디렉토리에 대한 원하는 위치를 입력합니다. 권장 디렉토리는 /u01/app/oraInventory입니다. oraInventory 디렉토리는 ORACLE\_BASE 디렉토리보다 한 레벨 위여야 합니다. 선택 리스트에서 oraInventory 그룹 이름을 선택합니다. 권장 이름은 oinstall입니다. Next 버튼을 눌러 설치를 계속합니다.

## 필수 구성 요소 검사 수행



ORACLE

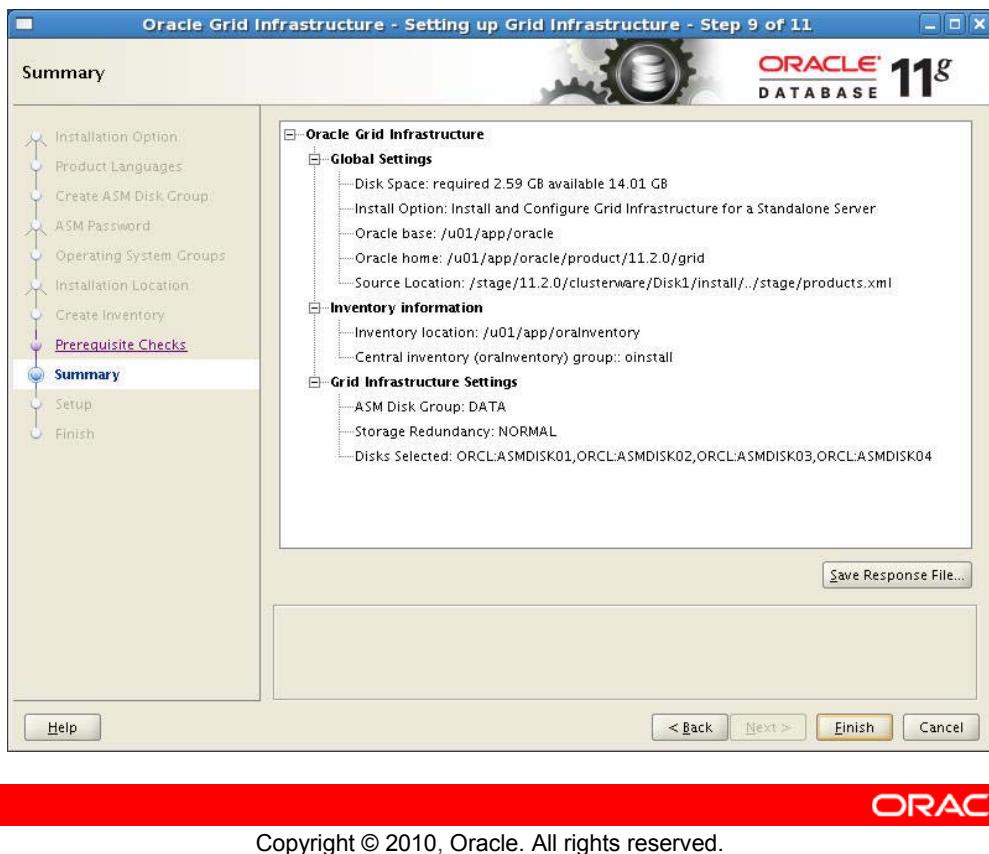
Copyright © 2010, Oracle. All rights reserved.

### 필수 구성 요소 검사 수행

"Perform Prerequisite Checks" 페이지가 나타납니다. 검사가 오류로 실패하지 않는 한 이 페이지에서는 수행할 작업이 없습니다. 오류가 발생하면 설치를 계속하기 전에 오류를 수정해야 합니다. 슬라이드 예에서는 필수 구성 요소 검사를 통과하지 못한 일부 커널 파라미터가 있습니다. 이러한 커널 파라미터를 수정하는 스크립트를 실행하고 검사를 다시 실행하라는 설명이 표시됩니다. 수정할 수 없는 오류가 있으면 문제를 수동으로 수정하거나 Ignore All을 선택하고 설치를 계속해야 합니다(실패한 검사가 중요하지 않은 경우).

**주:** OUI로 생성된 runfixup.sh 스크립트로 모든 Failure를 수정할 수는 없습니다. 그러한 경우 설치를 다시 시도하기 전에 수동으로 문제를 해결해야 합니다.

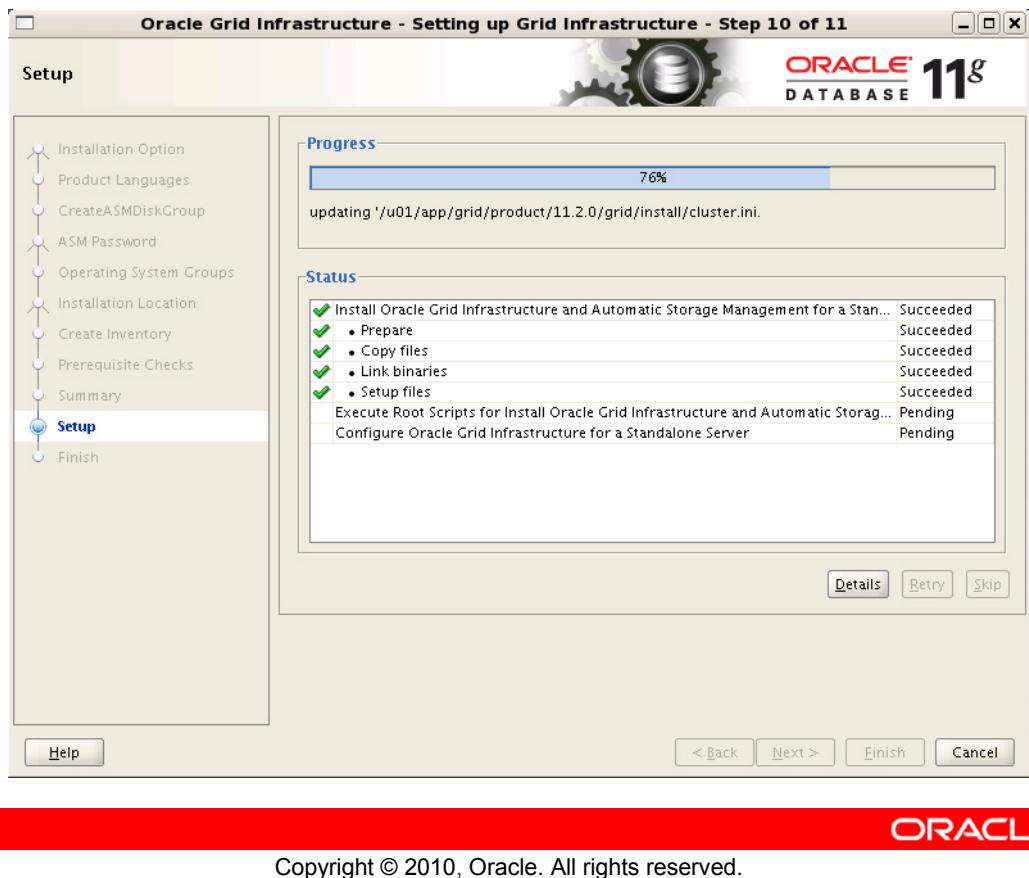
## 설치 요약 데이터 확인



### 설치 요약 데이터 확인

"Summary" 페이지가 표시됩니다. 이전 페이지에서 제공된 설치 대답과 관련하여 제공된 정보를 검토합니다. 대화식 설치 질문 및 대답을 응답 파일로 저장할 수 있는 옵션이 제공됩니다. 응답 파일은 이후 설치 시 이 설치를 수행할 때와 같은 방식으로 자동 설치를 수행하는 데 사용할 수 있습니다. 응답 파일은 텍스트 파일이며, 텍스트 편집 도구를 사용하여 다른 설치에 사용할 수 있도록 수정할 수 있습니다. Finish 버튼을 눌러 설치를 계속합니다.

## 설치 진행 상황 모니터



### 설치 진행 상황 모니터

설치 진행 상황을 보여주는 "Setup" 페이지가 나타납니다. 진행 상황에는 설치를 위한 준비, 파일 복사, 라이브러리 링크 및 설치 파일 생성이 포함됩니다. 설치가 오류로 실패하지 않는 한 이 페이지에서는 수행할 작업이 없습니다. 오류가 발생하면 설치를 계속하기 전에 오류를 수정해야 합니다.

## root 구성 스크립트 실행



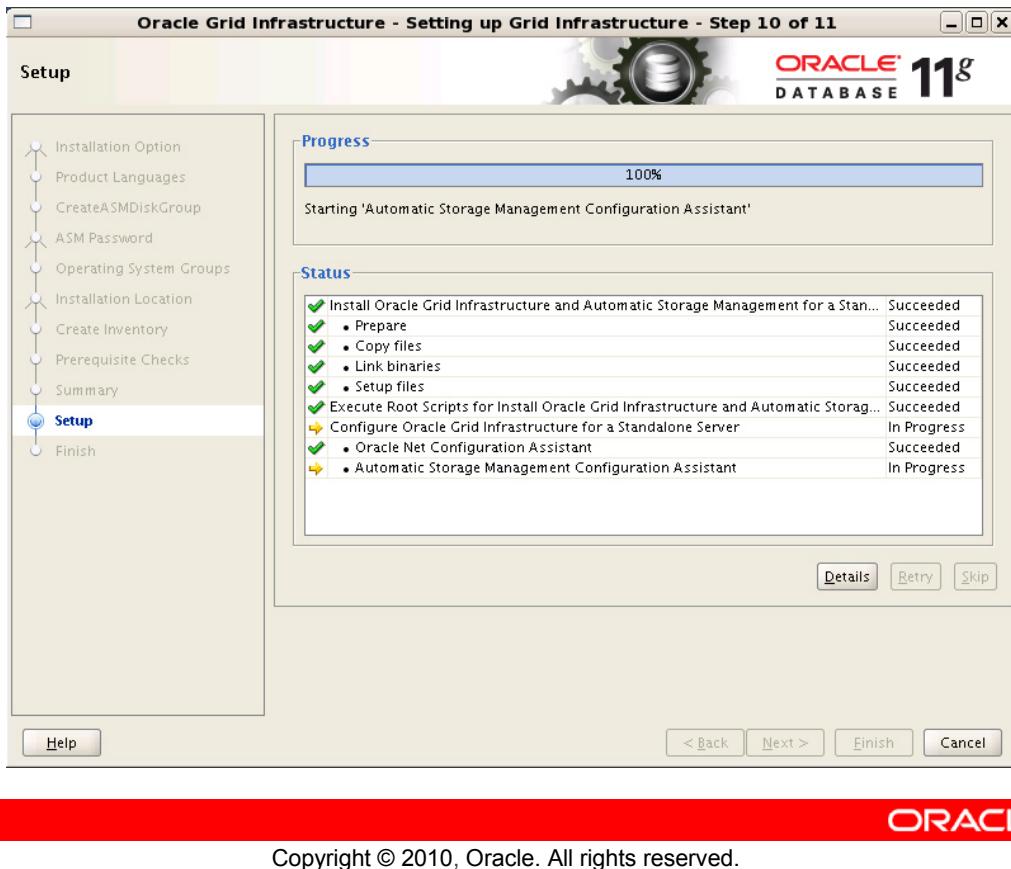
ORACLE

Copyright © 2010, Oracle. All rights reserved.

### root 구성 스크립트 실행

root 유저로 실행되어야 하는 구성 스크립트를 표시하는 "Execute Configuration Scripts" 대화상자 페이지가 옆에 표시됩니다. orainstRoot.sh 스크립트는 읽기, 쓰기 및 실행 권한을 완전히 제거하는 것과 함께 그룹에 대한 읽기 및 쓰기 권한을 추가하여 Oracle Central Inventory 디렉토리의 권한을 변경합니다. root.sh 스크립트는 파일을 /usr/local/bin 디렉토리에 복사하고, /etc/oratab 파일을 생성하고, grid 유저에 대한 OCR 키를 생성하고, ohasd daemon을 시작하고, /etc/inittab를 수정하여 시스템이 시작될 때 ohasd daemon 을 자동으로 시작합니다. root 스크립트를 실행한 다음에는 OK 버튼을 눌러 "Setup" 진행 상황 페이지로 돌아와서 설치를 계속합니다.

# Configuration Assistants 실행



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Configuration Assistants 실행

Oracle Net Configuration Assistant(`netca`)는 자동 설치 방법으로 호출되어 네트워크 구성 파일을 생성합니다. 그 다음에는 Automatic Storage Management Configuration Assistant(`asmca`)가 ASM 디스크 그룹을 생성하고 ASM 구성 요소를 Oracle Restart에 등록합니다.

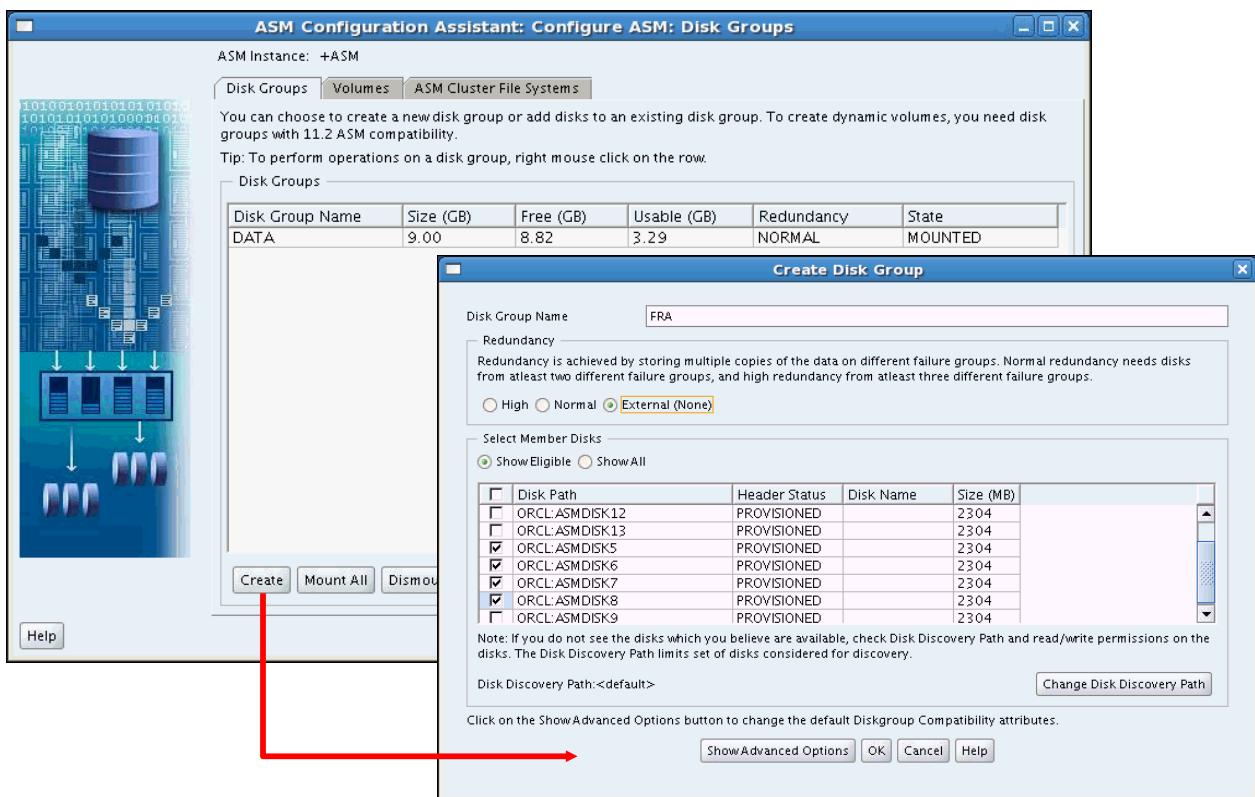
## 설치 완료



### 설치 완료

설치가 끝나면 "Finish" 페이지가 표시됩니다. Close 버튼을 눌러 OUI 유트리티를 종료합니다.

## FRA 디스크 그룹 구성



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### FRA 디스크 그룹 구성

Oracle 그리드 Infrastructure 설치 중에는 디스크 그룹을 하나만 구성할 수 있기 때문에 추가 디스크 그룹 생성은 수동으로 수행해야 합니다. 이 시나리오에서는 데이터베이스에 대한 Fast Recovery Area용으로 사용할 수 있는 FRA 디스크 그룹이 필요합니다. ASM Configuration Assistant(asmca) 유ти리티는 새 ASM 디스크 그룹을 쉽게 생성하고 기존 ASM 디스크 그룹을 제거할 수 있는 직관적인 GUI 인터페이스를 제공합니다.

## 퀴즈

**Universal Installer는 Oracle 소프트웨어를 설치하기 위한 모든 구성은 수행합니다.**

- 1. 참**
- 2. 거짓**

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

**해답: 2**

**퀴즈 설명**

예를 들어, UNIX와 Linux에서는 설치 전에 시스템 유저 및 그룹을 생성해야 합니다.

## 퀴즈

그리드 Infrastructure 설치 중에는 다음을 수행할 수 있습니다.

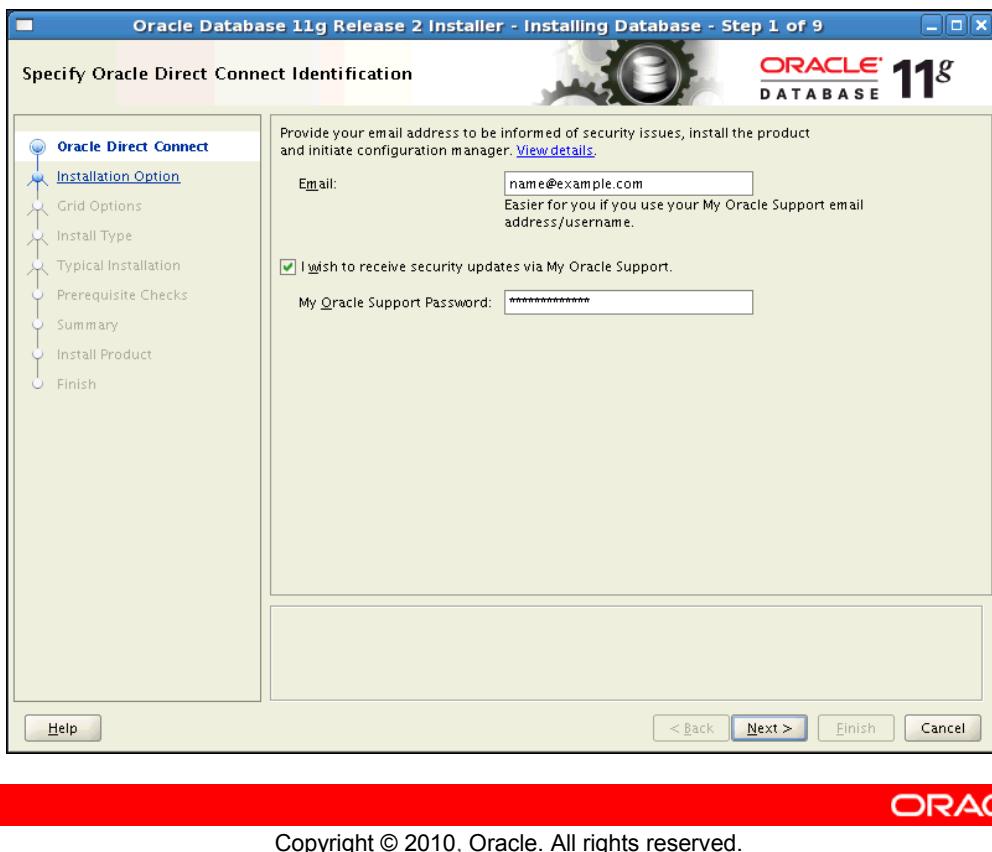
1. ASM Instance에 대한 정확한 데이터 파일 위치 지정
2. 하나의 DISKGROUP만 생성
3. ASM Instance에 대한 SGA 크기 지정
4. 여러 DISKGROUPS 생성



Copyright © 2010, Oracle. All rights reserved.

해답: 2

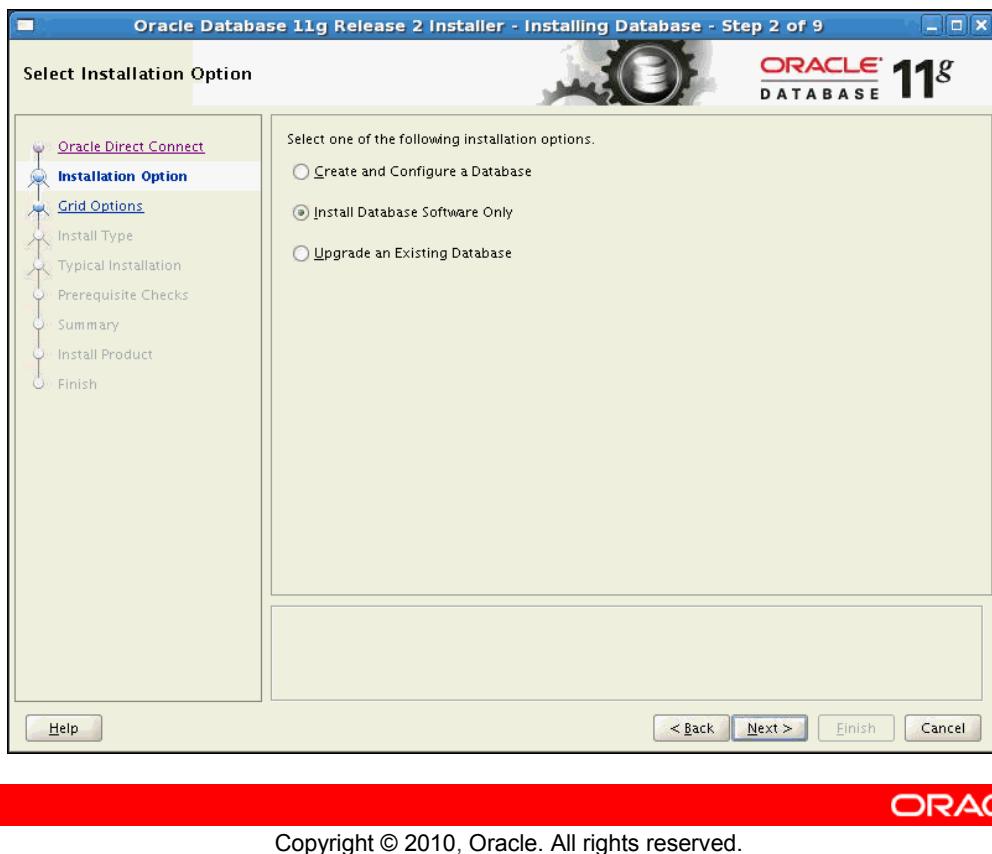
## 2부: 오라클 데이터베이스 소프트웨어 설치



### 2부: 오라클 데이터베이스 소프트웨어 설치

이제 오라클 데이터베이스 소프트웨어에 대한 설치 단계를 설명하겠습니다. 앞에서와 마찬가지로 Oracle 소프트웨어를 설치하고 데이터베이스를 생성 및 관리할 수 있는 관리 그룹의 멤버로 컴퓨터에 로그온합니다. 데이터베이스 배포 CD를 CD 드라이브에 넣거나 오라클 데이터베이스 스테이지 위치로 이동합니다. 이 위치에서 `./runInstaller`를 입력하여 OUI(Oracle Universal Installer)를 시작합니다. 필요한 경우 보안 문제에 대한 알림을 받을 전자 메일 주소를 입력합니다. My Oracle Support를 통해 보안 갱신을 받으려면 My Oracle Support 암호도 포함시킵니다. 전자 메일 주소를 입력하지 않으면 구성상의 중요 문제에 대한 알림을 받지 않을지 확인하는 경고 메시지가 표시됩니다. 이 경고에 대해 Yes를 누르면 설치가 계속됩니다.

## 설치 유형 선택

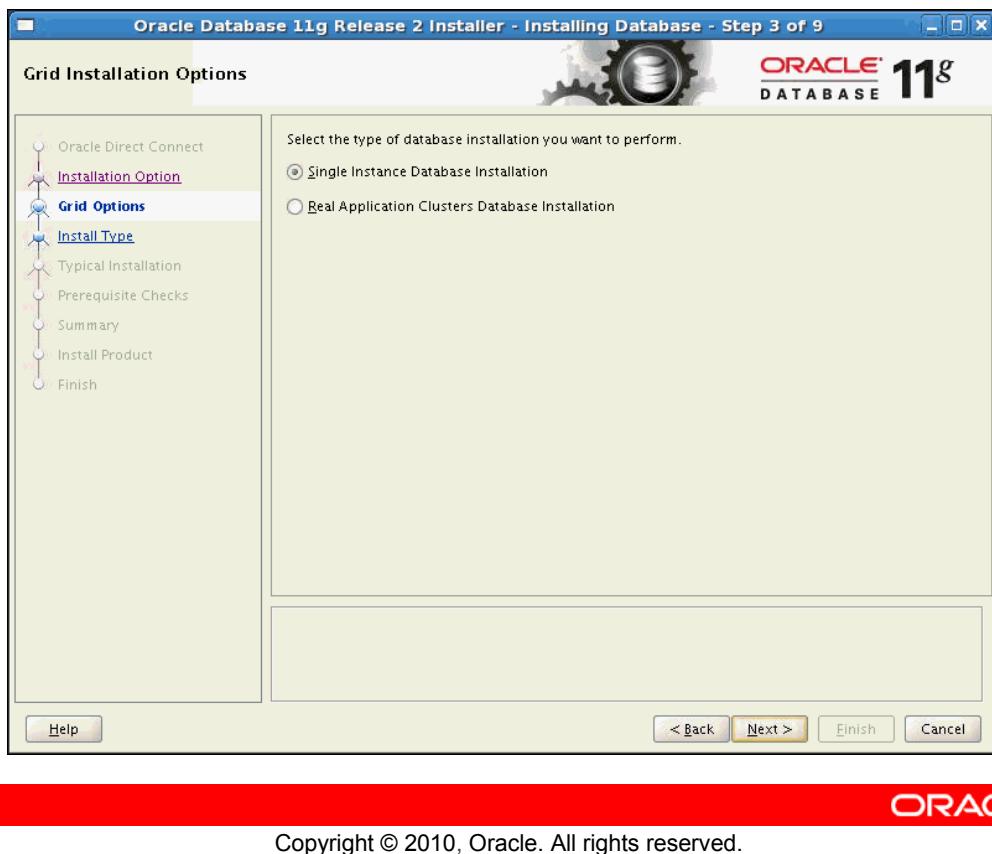


### 설치 유형 선택

Select Installation Option 페이지가 표시됩니다. 수행하려는 설치 유형을 선택하고 Next를 누릅니다.

- **Create and Configure a Database:** 이 옵션은 제품 설치 후 데이터베이스를 생성합니다.
- **Install Database Software Only:** 이 옵션은 오라클 데이터베이스 이전만 설치합니다.
- **Upgrade an Existing Database:** 이 옵션은 이전 릴리스의 데이터베이스를 업그레이드하는 데 사용됩니다.

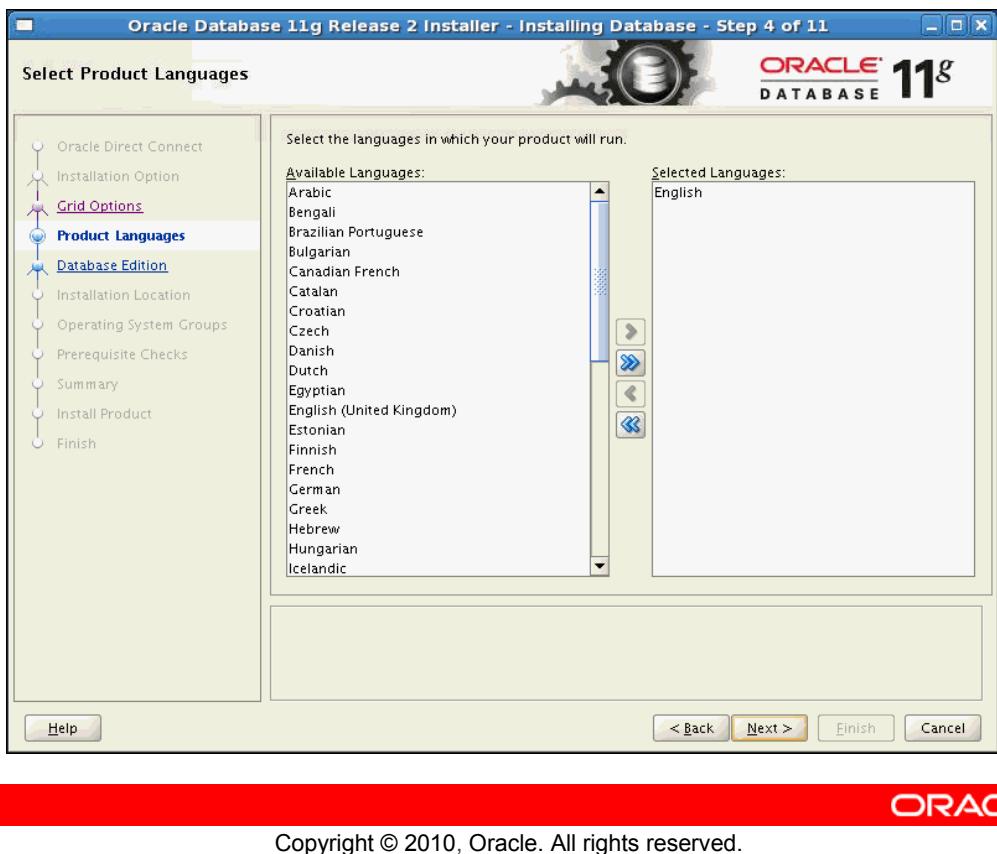
## 그리드 설치 옵션 선택



### 그리드 설치 옵션 선택

Grid Installation Option 페이지가 표시됩니다. 단일 Instance 데이터베이스 설치인지 또는 클러스터에 Oracle RAC 데이터베이스 설치인지 선택합니다. Next를 눌러 계속합니다.

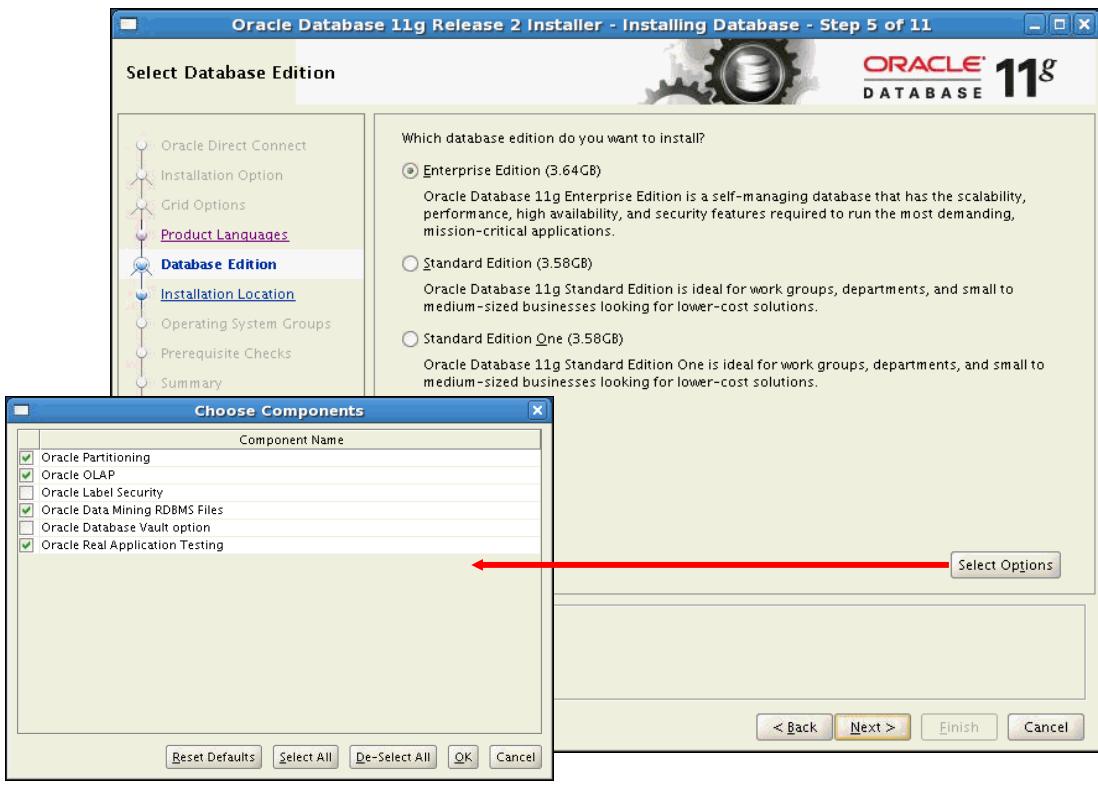
## 언어 설정 선택



### 언어 설정 선택

Select Product Languages 페이지가 표시됩니다. 여기에서는 제품을 실행할 모든 언어를 선택합니다. English는 기본적으로 선택되어 있으며 Selected Languages list에서 제거할 수 없습니다. Next를 눌러 계속합니다.

## 데이터베이스 버전 선택



Copyright © 2010, Oracle. All rights reserved.

ORACLE

### 데이터베이스 버전 선택

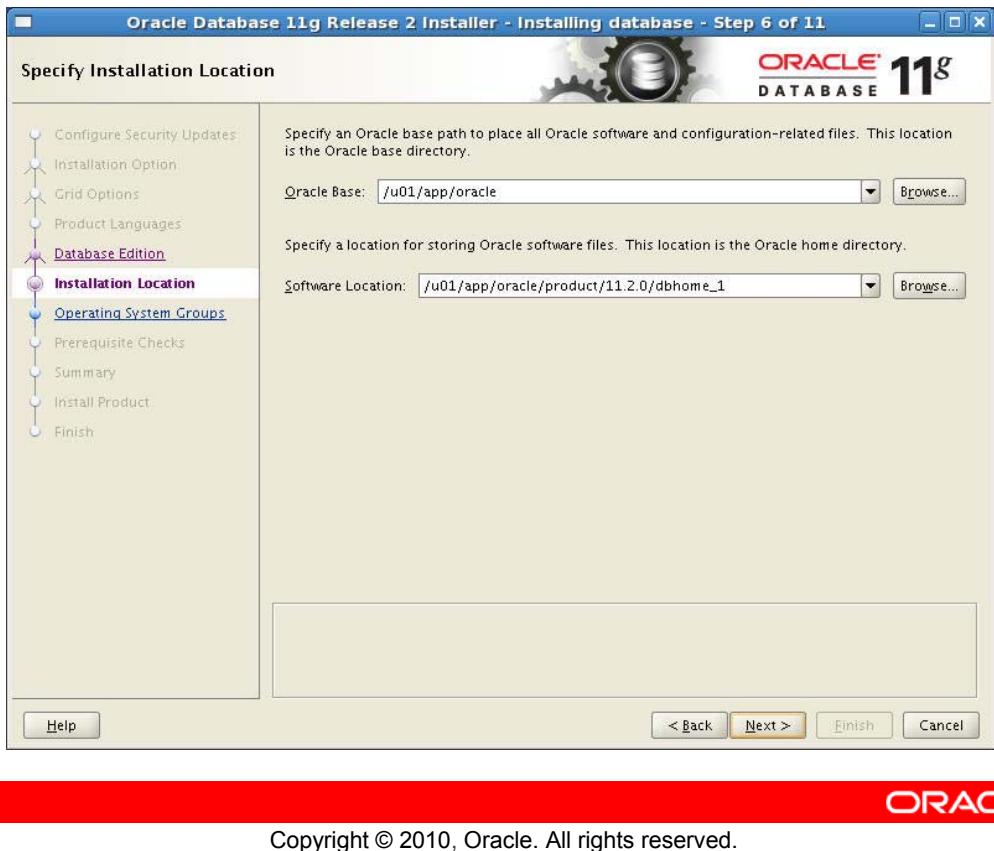
Select Database Edition 페이지가 표시됩니다. 다음 세 가지 버전 중에서 선택합니다.

- Enterprise Edition:** 이 버전은 선택한 클러스터화된 서버 또는 단일 서버에서 성능, 확장성, 보안 및 안정성을 제공하는 자체 관리 데이터베이스를 제공합니다. 이 버전은 가장 까다로운 트랜잭션 처리, business intelligence 및 컨텐트 관리 응용 프로그램을 쉽게 관리할 수 있는 포괄적인 기능들을 제공합니다.
- Standard Edition:** 이 버전은 최대 4개의 소켓까지 포함된 서버를 위한 완전한 기능의 데이터베이스를 제공합니다. 이 버전은 고가용성을 위한 Oracle Real Application Clusters를 포함하며, 엔터프라이즈급 성능 및 보안을 제공하고, 관리가 간편하며, 요구 증가에 따라 쉽게 확장할 수 있습니다. 또한 Enterprise Edition과의 상위 호환도 지원됩니다.
- Standard Edition One:** 이 버전은 최대 2개의 소켓까지 포함된 서버를 위한 완전한 기능의 데이터베이스를 제공합니다. 이 버전은 요구 증가에 따라 쉽게 확장할 수 있는 엔터프라이즈급 성능, 보안 및 관리 기능을 제공합니다. 또한 다른 데이터베이스 버전들과의 상위 호환도 지원됩니다.

설치할 구성 요소를 추가로 커스터마이즈하려면 Select Options 버튼을 누릅니다.

Select Database Edition 페이지에서 모든 항목을 선택한 다음에는 Next를 누릅니다.

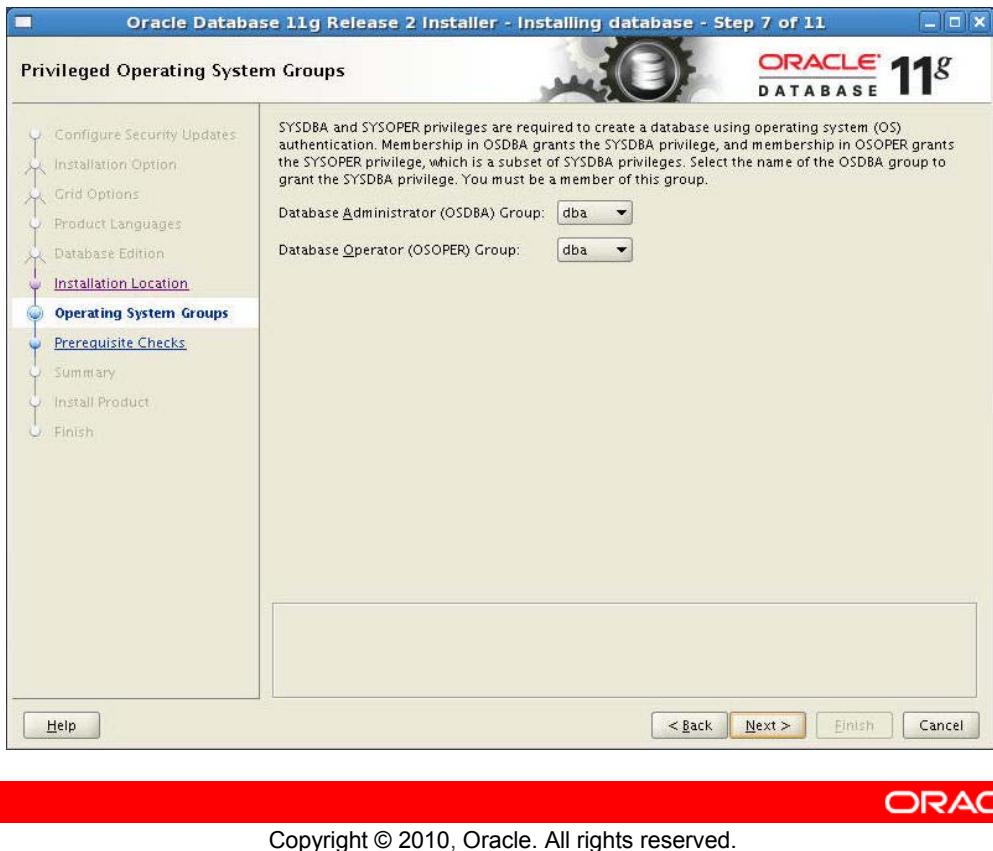
## 설치 위치 지정



### 설치 위치 지정

Installation Location 페이지가 옆에 표시됩니다. 권장되는 오라클 기본 경로는 기본적으로 표시됩니다. 유저의 요구에 따라 경로를 변경할 수 있습니다. Software Location 섹션에서 기본값을 사용하거나 오라클 구성 요소를 설치하려는 Oracle 홈 이름 및 디렉토리 경로를 입력할 수 있습니다. 디렉토리 경로에는 공백이 포함되지 않습니다. Next를 눌러 설치 프로세스를 계속합니다.

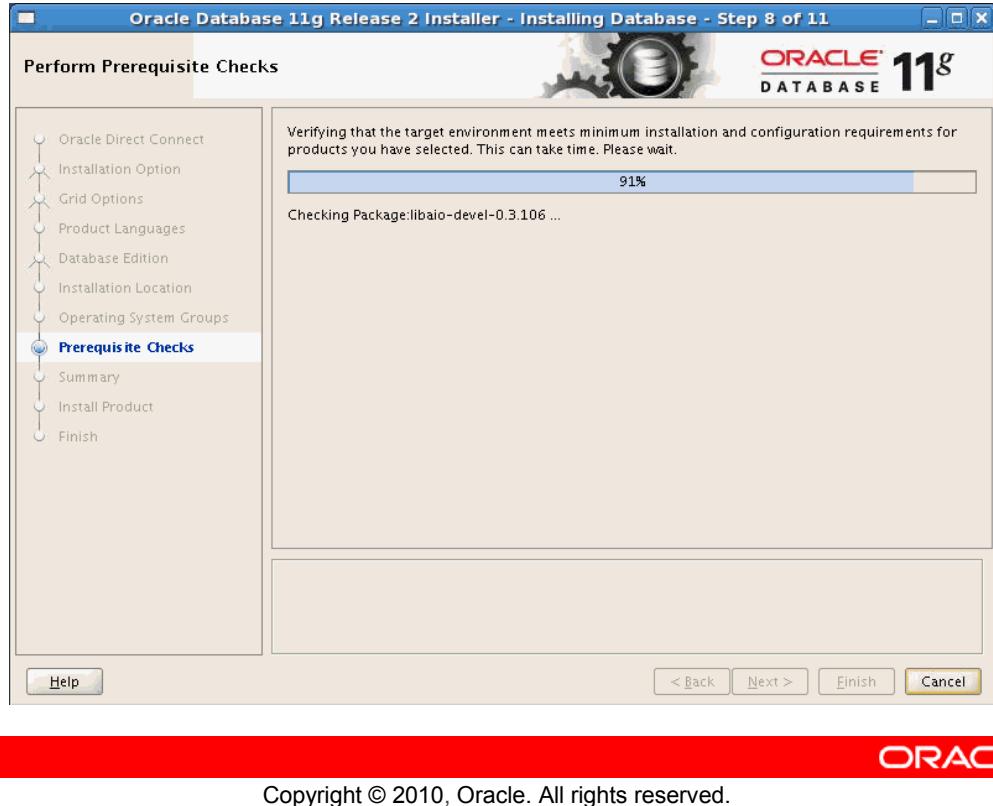
## 운영 체제 그룹 선택



### 운영 체제 그룹 선택

Privileged Operating System Groups 페이지가 표시됩니다. OSDBA 및 OSOPER 권한에 대한 적합한 운영 체제 그룹을 선택합니다. 기본적으로 dba는 OSDBA에 대해 지정되고 oper은 OSOPER에 대해 지정됩니다. 이 과정의 예제에서는 책임 구분을 구현하지 않는 독립형 설치를 수행 중이므로 두 개 모두에 대해 dba를 사용합니다. Next를 눌러 계속합니다.

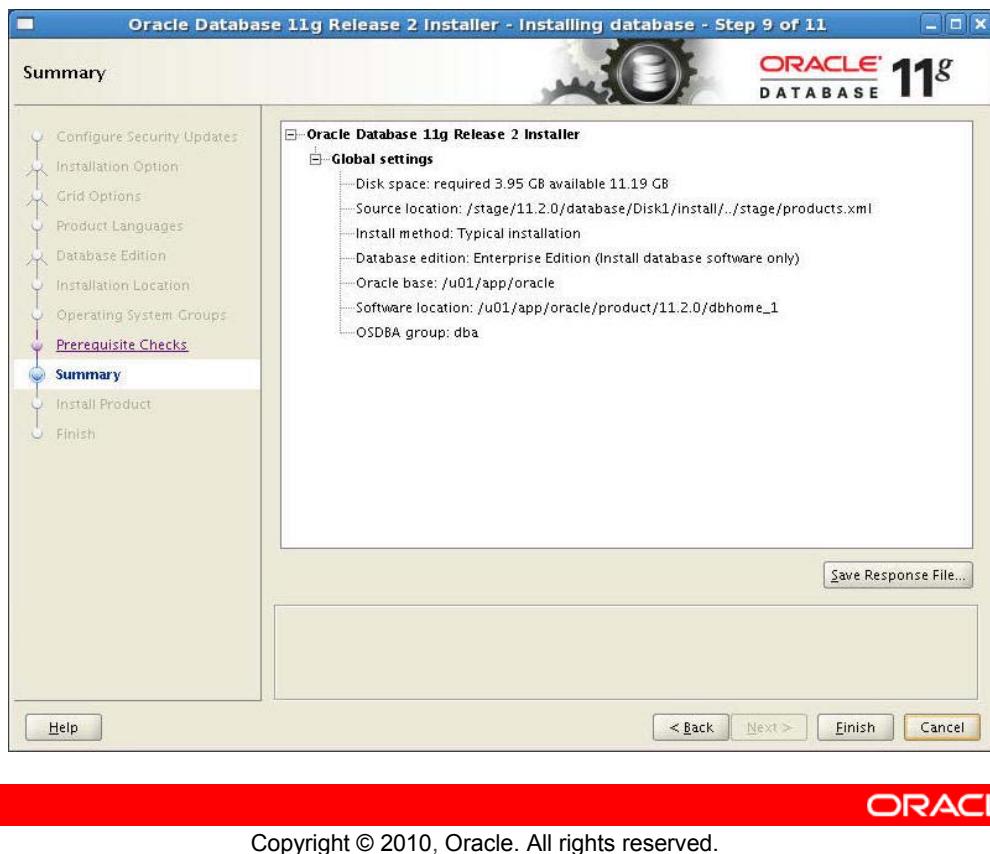
## 필수 구성 요소 검사 수행



### 필수 구성 요소 검사 수행

Perform Prerequisite Checks 페이지가 표시됩니다. OUI는 여러 필수 구성 요소 검사를 수행합니다. 검사가 100% 수행된 다음에는 테스트가 실패한 경우 OUI가 정보를 반환합니다. 검사가 오류로 실패하지 않는 한 이 페이지에서는 수행할 작업이 없습니다. 필수 구성 요소 검사가 실패하면 Fix & Check Again을 눌러서 OUI가 가능한 문제 해결을 위한 스크립트를 생성하는 페이지가 제공됩니다. OUI에서 수정할 수 없는 오류가 있으면 문제를 수동으로 수정하거나 Ignore All 을 선택하고 설치를 계속해야 합니다(실패한 검사가 중요하지 않은 경우).

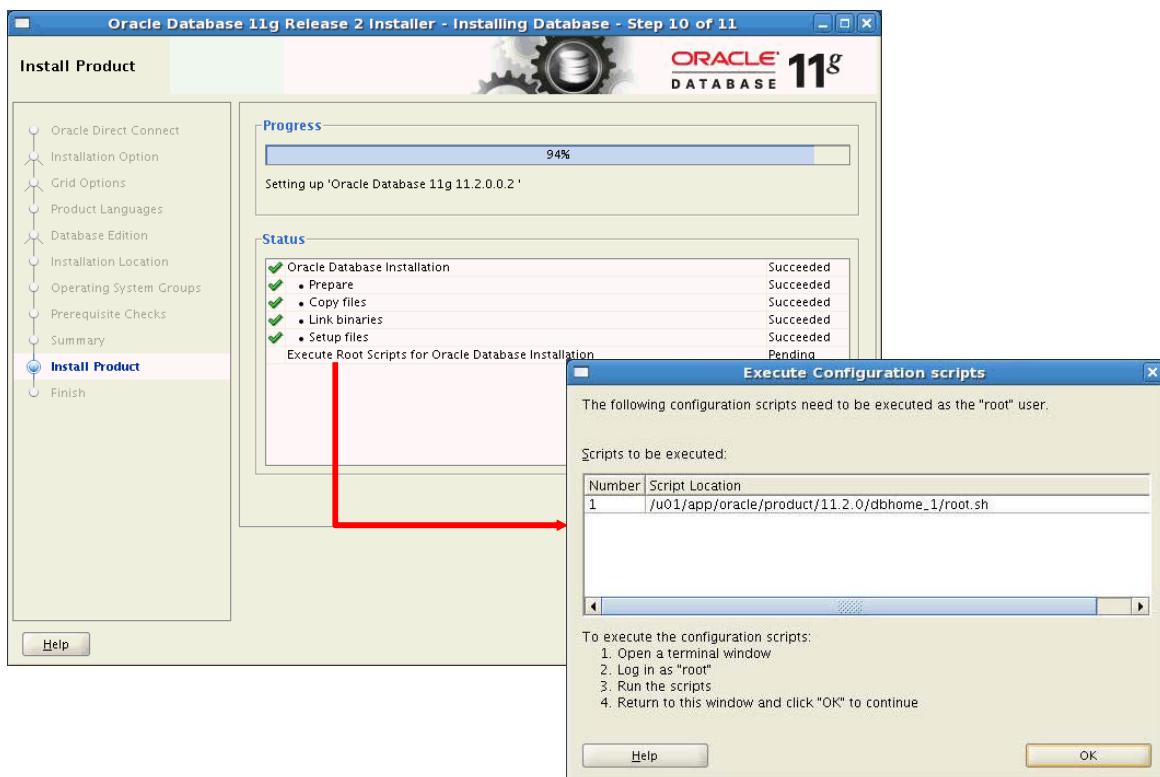
## 설치 요약 페이지



### 설치 요약 페이지

"Summary" 페이지가 표시됩니다. 이전 페이지에서 제공된 설치 대답과 관련하여 제공된 정보를 검토합니다. 대화식 설치 질문 및 대답을 응답 파일로 저장할 수 있는 옵션이 제공됩니다. 응답 파일은 이후 설치 시 이 설치를 수행할 때와 같은 방식으로 자동 설치를 수행하는 데 사용할 수 있습니다. 응답 파일은 텍스트 파일이며, 텍스트 편집 도구를 사용하여 다른 설치에 사용할 수 있도록 수정할 수 있습니다. Finish 버튼을 눌러 설치를 계속합니다.

## 제품 설치 페이지



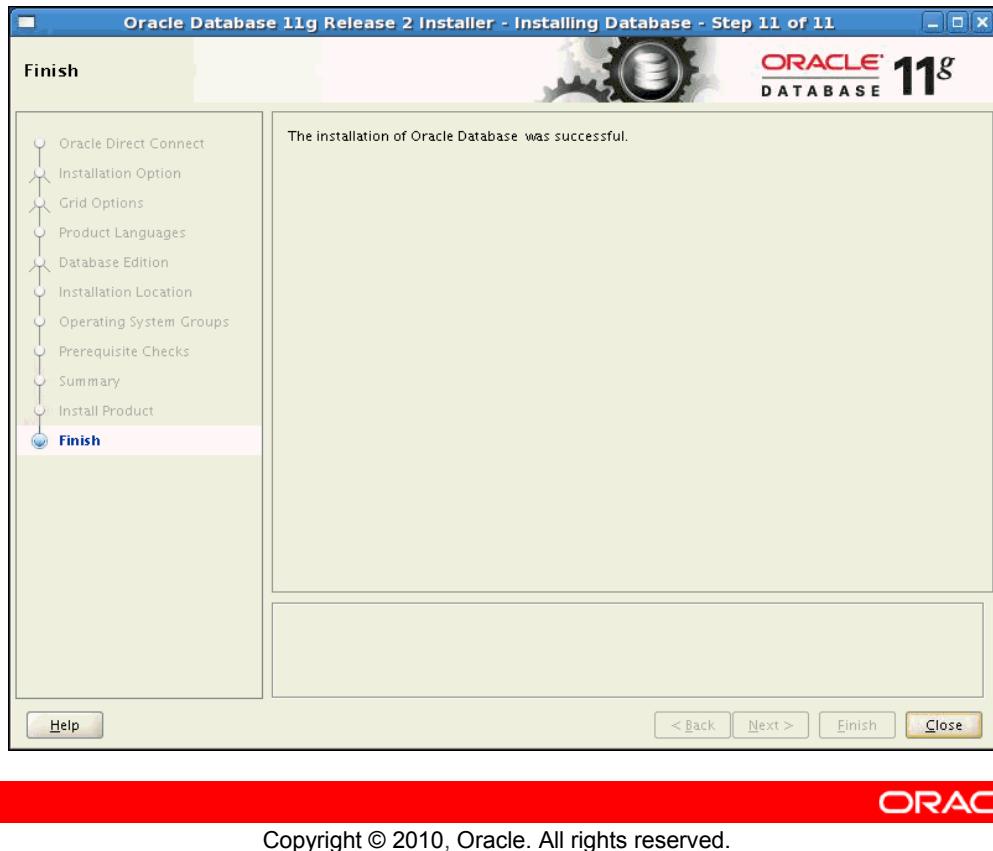
### 제품 설치 페이지

이 시점에서 설치 프로세스가 일시 정지되어 root 유저로 추가 구성 스크립트를 실행할 수 있습니다. 별개의 터미널 window에서 다음을 입력합니다.

```
$ su
# password: oracle<root 암호로, window에는 나타나지 않습니다.>
# /u01/app/oracle/product/11.2.0/dbhome_1/root.sh
```

Linux 또는 UNIX 설치 중에 로컬 bin 디렉토리에 대해 기본값을 적용합니다. 스크립트가 완료되면 root 계정을 나가고 window를 닫은 다음 Execute Configuration 스크립트 대화상자에서 OK를 눌러 설치가 완료되도록 합니다.

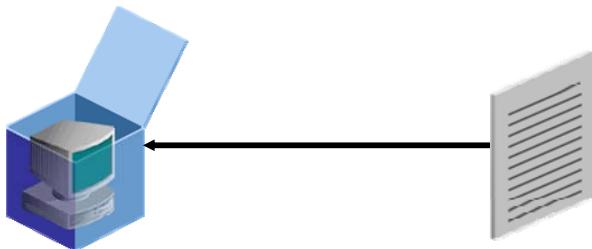
# 설치 완료



## 설치 완료

모든 설치 단계가 완료되면 "Finish" 페이지가 표시됩니다. Close 버튼을 눌러 OUI 유ти리티를 종료합니다.

## 설치 옵션: 자동 모드



**자동 모드에서 OUI를 사용하여 Oracle 제품을 설치하고 구성하려면 다음 단계를 수행하십시오.**

### 1. 응답 파일을 준비합니다.

- Oracle 소프트웨어에서 제공된 파일 템플리트 중 하나를 사용합니다.
- 요약 페이지에서 Save Response File을 눌러 OUI를 사용한 설치 프로세스 중 응답 파일을 기록합니다.

### 2. 자동 모드 또는 억제 모드에서 OUI를 실행합니다.

`./runInstaller -silent -responsefile <filename>`  
필요한 경우 NetCA 및 DBCA를 자동 모드로 실행합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 설치 옵션: 자동 모드

자동 모드 또는 억제 모드에서 OUI를 사용하여 Oracle 제품을 설치하고 구성하려면 다음을 수행하십시오.

1. 응답 파일을 준비합니다. 이 작업은 각 제품 및 설치 유형에 대해 제공되는 enterprise.rsp, standard.rsp 및 netca.rsp와 같은 파일 템플리트 중 하나를 사용하여 수행됩니다. 이러한 템플리트가 적합하지 않은 시나리오의 경우에는 대화식 모드에서 OUI를 사용하여 응답 파일을 기록한 다음 이를 편집하여 자동 모드나 억제 모드 설치를 완료하는 데 사용할 수 있습니다. 이렇게 하려면 OUI를 사용한 대화식 설치 중에 요약 페이지에서 Save Response File을 누릅니다.
2. 자동 모드 또는 억제 모드에서 OUI를 실행합니다. 설치가 끝나면 \$ORACLE\_BASE/oraInventory/orainstRoot.sh 및 \$ORACLE\_HOME/root.sh를 실행합니다. 소프트웨어 전용 설치를 완료한 경우 필요하면 자동 모드 또는 비대화식 모드에서 Oracle NetCA(Net Configuration Assistant) 및 DBCA(Database Configuration Assistant)를 실행합니다.

자세한 내용은 OS별 *Oracle Database Installation Guide*를 참조하십시오.

## 퀴즈

응답 파일의 특징은 다음과 같습니다.

- 1. 이진 편집 프로그램을 사용하여 편집할 수 있는 binary file입니다.**
- 2. 설치 프로그램에서 생성할 수 있는 binary file입니다.**
- 3. 편집할 수 없지만 설치 프로그램에서 생성할 수 있는 텍스트 파일입니다.**
- 4. 텍스트 편집기로 편집할 수 있는 텍스트 파일입니다.**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 4

## 퀴즈

데이터베이스 소프트웨어 설치 중에 지정할 수 있는 그룹은 무엇입니까?

1. **osoper 그룹**
2. **osasm 그룹**
3. **osdba 그룹**
4. **osadmin 그룹**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1, 3

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- DBA(데이터베이스 관리자)로서의 역할과 일반적인 작업 및 도구 설명
- 오라클 데이터베이스 설치 계획
- 독립형 서버를 위한 Oracle 그리드 Infrastructure 설치
- 오라클 데이터베이스 소프트웨어 설치



Copyright © 2010, Oracle. All rights reserved.

## 연습 2 개요: 데이터베이스 환경 준비

이 연습에서는 Oracle Universal Installer를 사용하여 Oracle 소프트웨어를 설치하는 과정을 다룹니다.

**주: 이 연습은 다음에 진행되는 모든 연습에서 중요한 역할을 하므로 반드시 마쳐야 합니다.**



Copyright © 2010, Oracle. All rights reserved.



# DBCA를 사용하여 오라클 데이터베이스 생성

3

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- DBCA(Database Configuration Assistant)를 사용하여 데이터베이스 생성
- DBCA를 사용하여 데이터베이스 생성 스크립트 작성
- DBCA를 사용하여 데이터베이스 설계 템플리트 관리
- DBCA를 사용하여 추가 작업 수행

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 데이터베이스 계획

**DBA는 다음과 같은 사항을 계획해야 합니다.**

- **데이터베이스의 논리적 저장 영역 구조 및 물리적 구현:**
  - 보유한 디스크 드라이브 수 및 사용 중인 저장 영역 유형
  - 필요한 데이터 파일 개수 (확장 계획)
  - 사용할 테이블스페이스 개수
  - 저장할 정보 유형
  - 유형 또는 크기에 따른 특별한 저장 영역 요구 사항
- **전반적인 데이터베이스 설계**
- **데이터베이스 백업 전략**



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 계획

데이터베이스의 논리적 저장 영역 구조가 시스템 성능 및 여러 데이터베이스 관리 작업에 미치는 영향에 대해 계획해야 합니다. 예를 들어, 데이터베이스의 테이블스페이스를 생성하기 전에 테이블스페이스를 구성할 데이터 파일의 개수, 각 테이블스페이스에 저장될 정보의 유형, 데이터 파일이 물리적으로 저장될 디스크 드라이브를 미리 결정해야 합니다. NAS(Network Attached Storage)의 가용성 및 전용(Private) 저장 영역 네트워크의 대역폭 같은 정보는 중요한 역할을 합니다. SAN(Storage Area Network)를 사용하려는 경우에는 논리적 볼륨이 구성되는 방식과 스트라이프 크기를 확인하는 것이 좋습니다.

데이터베이스 구조의 전체적인 논리적 저장 영역을 계획할 때는 데이터베이스가 실제로 생성되어 실행될 때 이 구조가 미치게 될 영향에 대해 고려하십시오. 유형 또는 크기로 인해 특별한 저장 영역이 필요한 데이터베이스 객체가 있을 수도 있습니다.

분산 데이터베이스 환경에서는 이 계획 단계가 특히 중요합니다. 자주 액세스하는 데이터의 물리적 위치에 따라 응용 프로그램의 성능이 크게 차이가 납니다.

계획 단계에서 데이터베이스의 백업 전략을 개발하십시오. 논리적 저장 영역 구조나 데이터베이스 설계를 변경하여 백업 효율성을 개선할 수 있습니다. 백업 전략은 뒤에 나오는 단원에서 소개됩니다.

## 데이터베이스: 예제

- **범용 또는 트랜잭션 처리:**
  - OLTP(온라인 트랜잭션 프로세싱) 시스템, 예: 소프트웨어 하우스 또는 보육원 등의 소매 대금 청구 시스템
- **커스텀:**
  - 다목적 데이터베이스(OLTP와 데이터 웨어하우스 기능의 조합)
- **데이터 웨어하우스:**
  - 조사 및 마케팅 데이터
  - 주세 또는 연방세 납부
  - 전문직 면허(의사, 간호사 등)

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스: 예제

데이터베이스는 유형에 따라 고유한 instance 및 저장 영역 요구 사항을 가지고 있습니다. 오라클 데이터베이스 소프트웨어에는 여러 가지 유형의 데이터베이스를 생성할 수 있는 템플리트가 포함되어 있습니다. 각 예의 특징은 다음과 같습니다.

- **범용:** 트랜잭션을 사용한 작업 및 중간 정도의 기간 동안 저장과 같은 범용 또는 트랜잭션 처리 용도
- **커스텀:** 범용 또는 데이터 웨어하우스 템플리트에 적합하지 않은 커스터마이즈된 데이터베이스 용도
- **데이터 웨어하우스:** 데이터를 장기간 저장하고 읽기 작업에서 해당 데이터를 검색합니다.

이 페이지와 이전 페이지의 정보는 DBA가 고려해야 할 사항에 대한 내용입니다. 본 과정은 전체적으로 이러한 사항을 해결하는 데 도움을 주기 위한 것입니다.

## 적합한 Character Set 선택

- 오라클 데이터베이스는 다양한 부류의 문자 인코딩 방식을 지원합니다.
  - 단일 바이트 Character Set
    - 7비트
    - 8비트
  - 유니코드를 포함한 멀티바이트 Character Set
- Character Set은 데이터베이스 생성 시에 선택합니다. Character Set은 나중에 변경하기가 어렵기 때문에 현재 및 이후의 업무 요구 사항을 가장 잘 충족시킬 수 있는 Character Set를 선택하십시오.
- 일반적으로는 가장 유연한 Character Set인 유니코드가 권장됩니다.

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 적합한 Character Set 선택

컴퓨터 시스템이 문자를 처리할 때 문자의 그래픽 표현 대신 숫자 코드를 사용합니다. 인코딩된 Character Set은 컴퓨터나 터미널이 표시하고 받을 수 있는 문자에 숫자 코드를 매핑합니다. 다양한 Character Set은 서로 다른 문자 범위를 지원합니다. Character Set은 일반적으로 특정 스크립트 작성을 기반으로 하기 때문에 여러 언어를 지원할 수 있습니다. 그러나 스크립트 기반 Character Set은 유사한 스크립트를 기반으로 한 언어 그룹만 지원하기 때문에 제한적입니다. Universal Character Set은 현재 대부분의 주요 스크립트를 포함하며 다국어 지원에 유용한 솔루션을 제공합니다. 유니코드 표준에 대한 내용은 <http://www.unicode.org> 웹 사이트를 참조하십시오.

오라클 데이터베이스는 단일 바이트, 가변 너비 멀티바이트 및 범용의 세 가지 종류의 인코딩 방식을 지원합니다. Character Set은 나중에 변경하기가 어렵기 때문에 현재 및 이후의 업무 요구 사항을 가장 잘 충족시킬 수 있는 올바른 Character Set를 선택하십시오. 최상의 성능을 위해 Character Set 변환을 피하고 원하는 언어에 대해 가장 효율적인 인코딩을 사용하는 Character Set를 선택합니다. 단일 바이트 Character Set은 멀티바이트 Character Set보다 성능이 높고 공간 요구 사항 측면에서도 가장 효율적입니다. 그러나 단일 바이트 Character Set은 지원할 수 있는 언어 개수에 제한이 있습니다. 올바른 데이터베이스 Character Set을 선택하려면 현재와 미래의 업무 요구 사항과 기술적 요구 사항(예: XML 및 Java 표준에는 유니코드가 필요)을 고려하십시오. 일반적으로 오라클은 모든 새 데이터베이스에 대해 유니코드 사용을 권장합니다. 유니코드가 가장 유연한 Character Set이며 이후에 변환할 필요가 없기 때문입니다.

## 적합한 Character Set 선택(계속)

### 단일 바이트 Character Set

단일 바이트 Character Set에서 각 문자는 1바이트를 차지합니다. 단일 바이트 인코딩 방식은 최대 128( $2^7$ )자까지 정의할 수 있고, 단일 바이트 인코딩 방식은 최대 256( $2^8$ )자까지 정의할 수 있습니다.

### 단일 바이트 방식의 예

#### 7비트 Character Set:

- ASCII(American Standard Code for Information Interchange) 7비트 미국 영어(US7ASCII)

#### 8비트 Character Set:

- ISO(International Organization for Standards) 8859-1 서유럽어(WE8ISO8859P1)
- DEC 8비트 서유럽어(WE8DEC)
- EBCDIC(Extended Binary Coded Decimal Interchange Code) Code Page 1144 8비트 이탈리아어(I8EBCDIC1144)

### 멀티바이트 Character Set

가변 너비 멀티바이트 Character Set는 문자당 하나 이상의 바이트로 표현됩니다. 멀티바이트 Character Set는 일반적으로 아시아 언어 지원에 사용됩니다. 일부 멀티바이트 인코딩 방식은 MSB(최상위 비트) 값을 사용하여 한 바이트가 단일 바이트를 나타내는지 아니면 한 문자를 구성하는 일련의 바이트에 속하는 것인지 나타냅니다. 그러나 다른 문자 인코딩 방식에서는 단일 바이트가 멀티바이트 문자와 구별됩니다. 장치에서 보내온 시프트 아웃 제어 코드는 시프트 인 코드가 발생할 때까지 연속 바이트가 2바이트 문자임을 나타냅니다. Shift 인코딩 방식은 주로 IBM 플랫폼에서 사용됩니다.

유니코드는 어떤 언어로 된 정보든 단일 Character Set를 사용하여 저장할 수 있는 범용 인코딩된 Character Set입니다. 유니코드는 플랫폼, 프로그램 또는 언어에 관계없이 모든 문자에 고유 코드 값을 제공합니다.

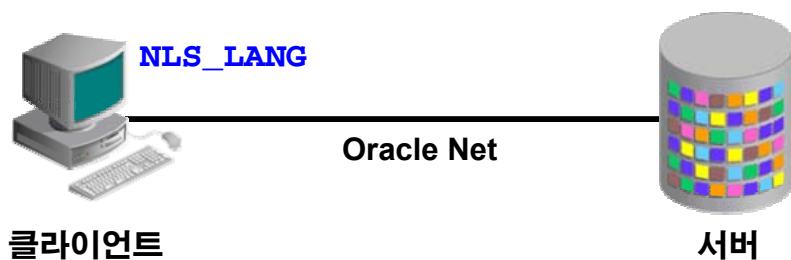
많은 소프트웨어와 하드웨어 공급업체가 유니코드 표준을 채택했으며 많은 운영 체제와 브라우저가 현재 유니코드를 지원합니다. 유니코드는 XML, Java, JavaScript, LDAP 및 WML과 같은 표준에 필요합니다. 또한 유니코드는 ISO/IEC 10646 표준과 동기화됩니다.

### 가변 너비 멀티바이트 방식의 예

- Shift-JIS 16비트 일본어(JA16SJIS)
- MS Windows Code Page 950 with Hong Kong Supplementary Character Set HKSCS-2001(ZHT16HKSCS)
- Unicode 4.0 UTF-8 Universal character set(AL32UTF8) - 가변 너비 유형의 인코딩이면서 ASCII의 절대 대집합입니다.
- 유니코드(AL16UTF16) – Microsoft Windows 2000 및 Windows XP에 사용되는 16비트 유니코드 인코딩입니다.

## Character Set 사용 방법

- Oracle Net는 클라이언트 NLS\_LANG 설정을 서버의 Character Set와 비교합니다.
- 필요한 경우 자동으로 투명하게 변환됩니다.



ORACLE®

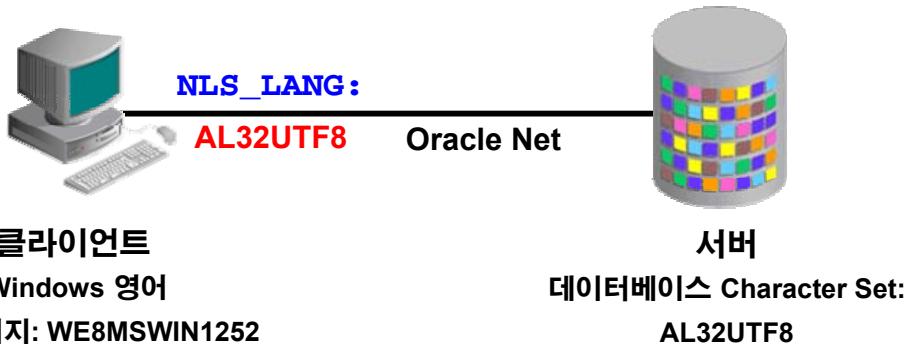
Copyright © 2010, Oracle. All rights reserved.

### Character Set 사용 방법

NLS\_LANG 파라미터는 클라이언트 터미널의 문자 인코딩 방식을 정의합니다. 클라이언트마다 다른 인코딩 방식을 사용할 수 있습니다. 클라이언트와 서버 간에 전달된 데이터는 두 인코딩 방식 간에 자동으로 변환됩니다. 데이터베이스의 인코딩 방식은 모든 클라이언트 인코딩 방식의 대집합이거나 이와 동등해야 합니다. 변환은 클라이언트 응용 프로그램에 대해 투명합니다. 데이터베이스 Character Set과 클라이언트 Character Set가 동일하면 데이터베이스는 전송되거나 수신되는 데이터의 Character Set가 동일하다고 가정하므로 검증이나 변환이 수행되지 않습니다. 클라이언트/서버 환경에서 클라이언트 응용 프로그램이 서버가 아닌 다른 플랫폼에 있고 해당 플랫폼이 동일한 문자 인코딩 방식을 사용하지 않는 경우 Character Set 변환이 필요할 수도 있습니다. 클라이언트와 서버 간에 전달된 문자 데이터는 두 인코딩 방식 간에 변환되어야 합니다. 문자 변환은 Oracle Net를 통해 자동으로 투명하게 발생합니다.

## 방지해야 할 문제

예제:



**변환이 필요하지 않은 것으로 보여 어떠한 변환도 발생하지 않습니다.**

**문제점: 데이터베이스에 잘못된 데이터가 입력됩니다.**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

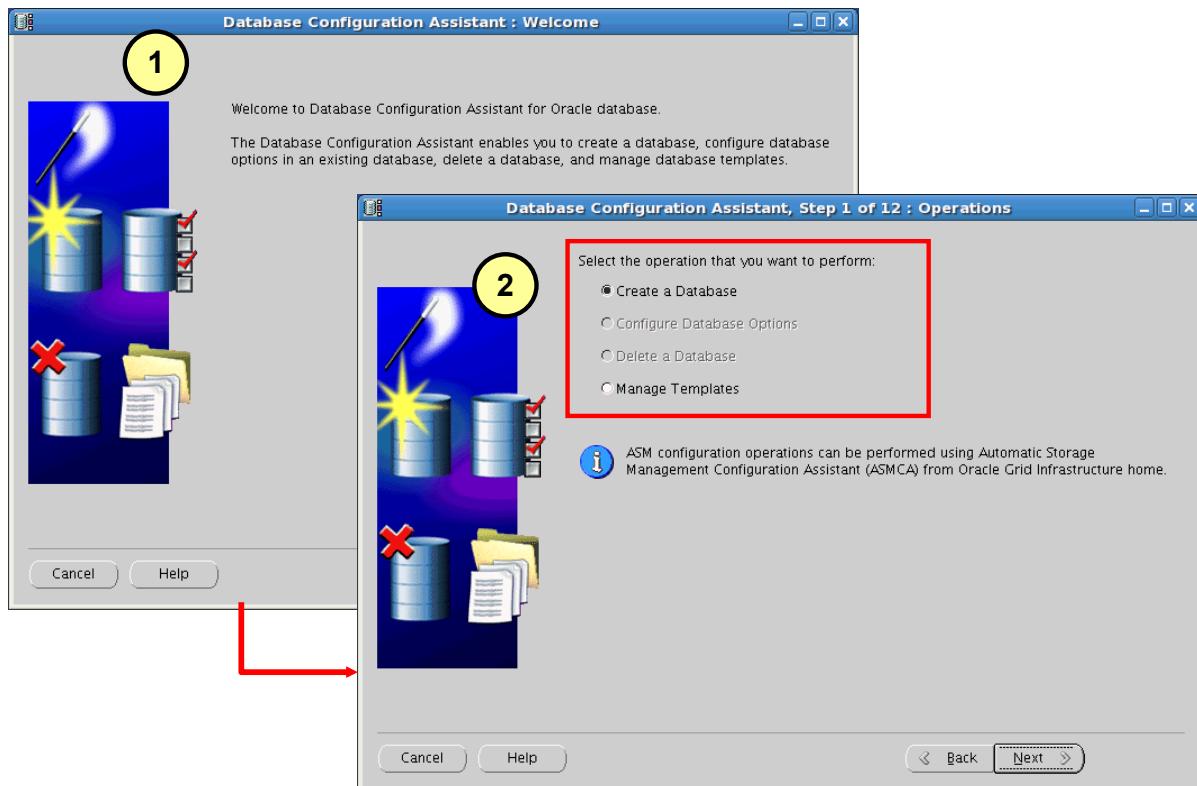
### 방지해야 할 문제

잘못된 데이터는 대개 클라이언트에서 NLS\_LANG 파라미터가 올바로 설정되지 않은 경우에 데이터베이스에 입력됩니다. NLS\_LANG 값은 들어오는 데이터의 인코딩을 반영해야 합니다.

- NLS\_LANG 파라미터가 올바로 설정된 경우 데이터베이스가 클라이언트 운영 체제로부터 들어오는 데이터를 자동으로 변환할 수 있습니다.
- NLS\_LANG 파라미터가 올바로 설정되지 않은 경우 데이터베이스에 입력되는 데이터가 올바로 변환되지 않습니다.

예를 들어, 데이터베이스 Character Set가 AL32UTF8이고 클라이언트가 영어 Windows 운영 체제 (코드 페이지: WE8MSWIN1252)를 사용하고 클라이언트의 NLS\_LANG 설정이 AL32UTF8이라고 가정해 보겠습니다. 이 경우 클라이언트의 NLS\_LANG 설정이 데이터베이스 Character Set와 일치하지 않기 때문에 데이터베이스에 입력되는 데이터는 WE8MSWIN1252로 인코딩되고 AL32UTF8 데이터로 변환되지 않습니다. 따라서 오라클 데이터베이스는 변환이 필요하지 않은 것으로 간주하고 데이터베이스에 잘못된 데이터가 입력됩니다.

# DBCA(Database Configuration Assistant)



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

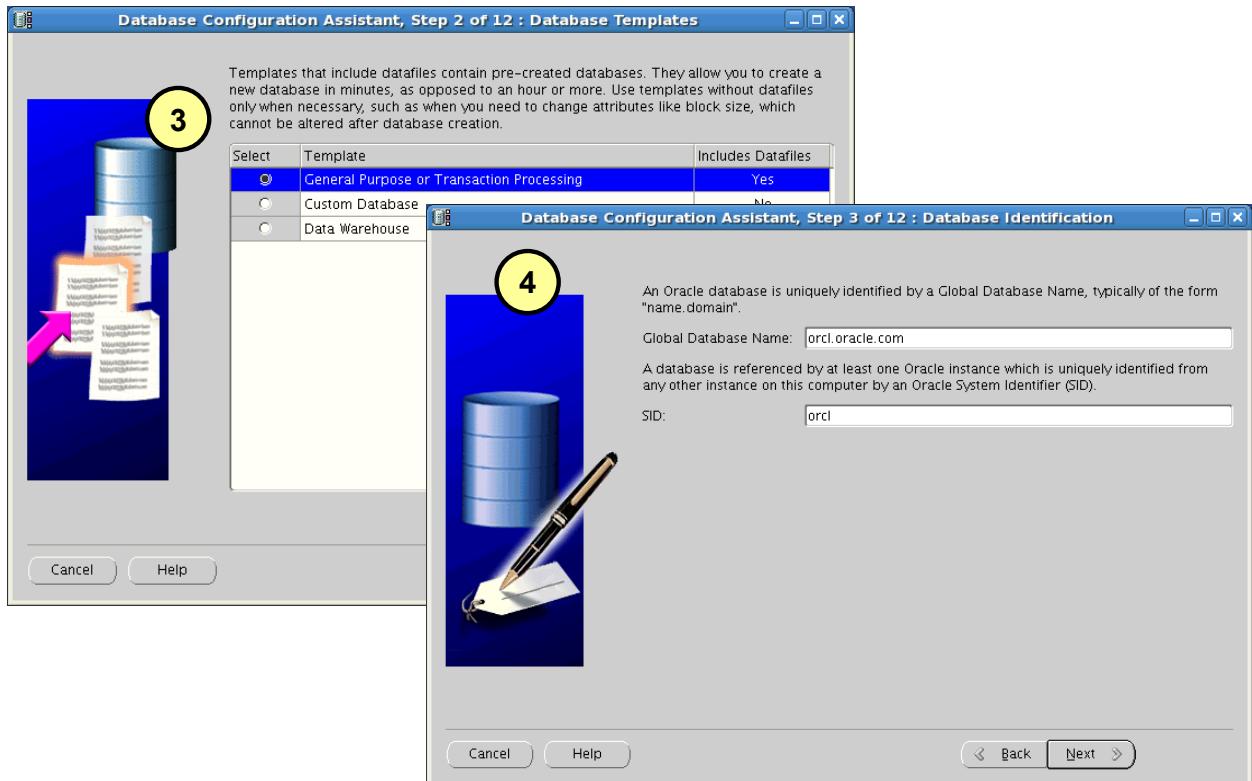
## DBCA(Database Configuration Assistant)

DBCA(Database Configuration Assistant)는 여러 작업들을 지원하는 몇 가지 선택 사항을 제공합니다. DBCA를 사용하여 데이터베이스를 생성 또는 삭제하거나 구성을 변경할 수 있습니다. 미리 정의된 템플릿 리스트에서 데이터베이스를 생성하거나 기존 데이터베이스를 예제로 사용하여 새 데이터베이스 또는 템플릿을 생성할 수 있습니다. DBCA는 유저 요구를 충족시키는 데이터베이스를 생성할 수 있게 해주는 몇 가지 옵션을 제공합니다. DBCA는 구성 정보를 입력하는 일련의 페이지를 제공합니다. 대부분의 페이지에서 DBCA는 유저가 수락하여 적용할 수 있는 기본 설정을 제공합니다.

DBCA를 사용하여 데이터베이스를 생성하는 것과 관련된 단계는 다음과 같습니다.

1. Oracle 소프트웨어를 설치할 수 있는 OS DBA 그룹의 멤버로 컴퓨터에 로그온합니다. 필요한 경우 환경 변수를 설정하고 dbca를 입력하여 DBCA를 호출합니다. 기본 Welcome 페이지가 표시되고 Next를 누르면 계속됩니다.
2. 수행할 작업을 선택합니다. 이 예에서는 Create a Database를 선택하고 Next를 눌러 작업을 시작합니다.

## DBCA를 사용하여 데이터베이스 생성



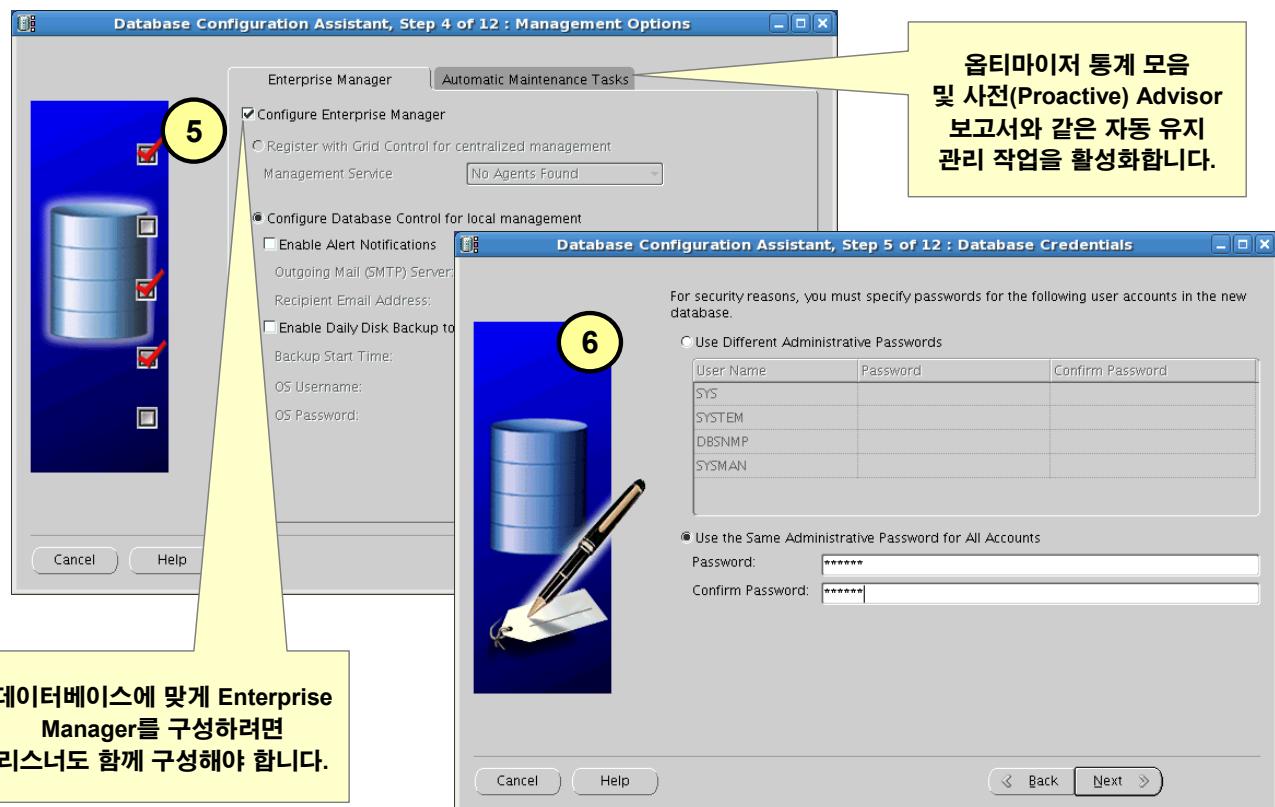
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## DBCA를 사용하여 데이터베이스 생성

3. **Database Templates:** 데이터베이스를 생성할 때 사용할 데이터베이스 템플리트 유형을 선택합니다. 데이터 파일을 포함하여 미리 구성된 데이터베이스를 복사하는 데이터베이스 템플리트는 범용 또는 트랜잭션 처리, 커스텀 데이터베이스 및 데이터 웨어하우스의 세 가지가 있습니다. 이러한 파일에는 콘트롤 파일, 리두 로그 파일 및 포함된 여러 테이블스페이스에 대한 데이터 파일이 포함됩니다. 각 데이터베이스 유형에 대한 구성을 보려면 Show Details를 누릅니다.  
보다 복잡한 환경의 경우 Custom Database 옵션을 선택할 수 있습니다.
4. **Database Identification:** `database_name.domain_name` 형식으로 Global Database Name을 입력하고 SID(시스템 식별자)도 입력합니다. SID는 기본적으로 데이터베이스 이름으로 설정되어 있으며 데이터베이스와 연관된 Instance를 고유하게 식별합니다.

## DBCA를 사용하여 데이터베이스 생성



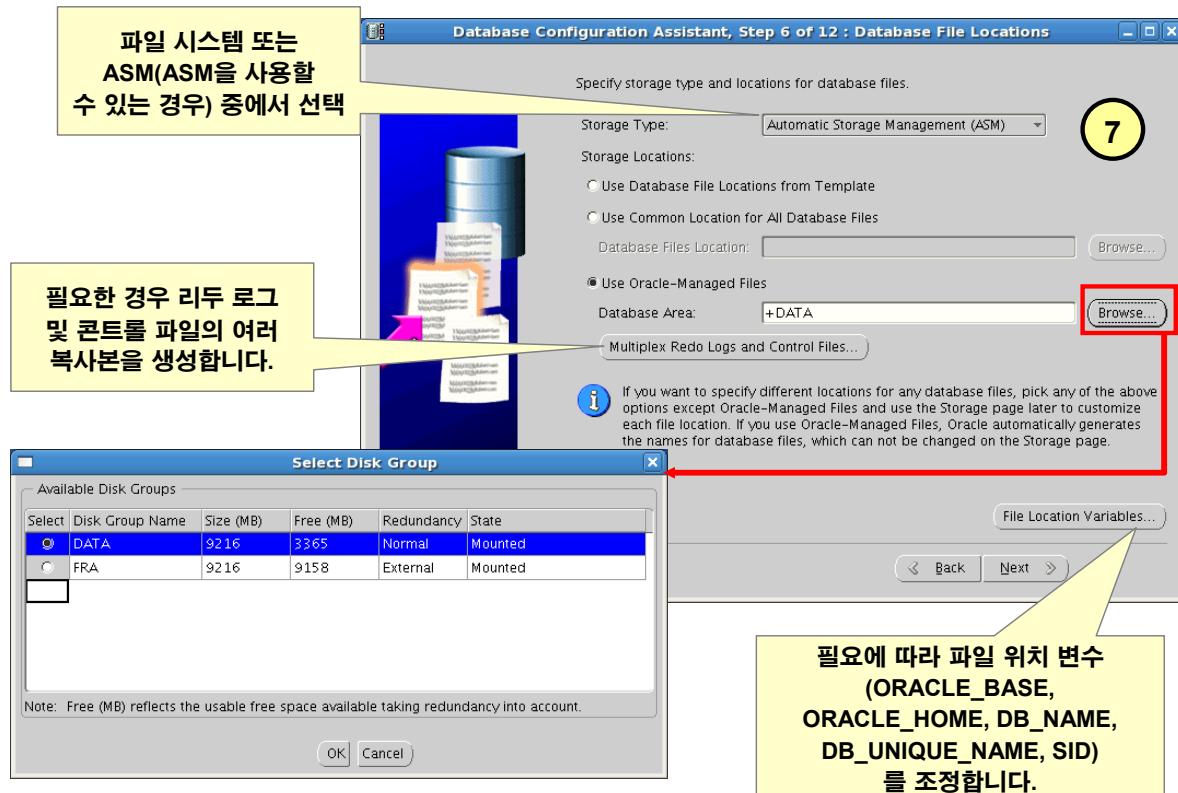
ORACLE

Copyright © 2010, Oracle. All rights reserved.

### DBCA를 사용하여 데이터베이스 생성(계속)

- Management Options:** 이 페이지에서 Oracle Enterprise Manager를 사용하여 데이터 베이스를 관리할 수 있도록 설정합니다. 기본값인 "Configure the Database with Enterprise Manager"를 선택합니다. 선택적으로 이 페이지에서 경고 알림 및 일별 디스크 백업 영역 설정을 구성할 수 있습니다. DBCA가 통신할 Enterprise Manager Grid Control 에이전트를 찾을 수 없는 경우 Register with Grid Control 옵션을 사용할 수 없습니다.  
**주:** 데이터베이스에 맞게 Enterprise Manager를 구성하려면 리스너도 함께 구성해야 합니다. 강의실에서는 리스너가 Oracle Grid Infrastructure 설치 중에 구성되어 있습니다.
- Database Credentials:** 이 페이지에서 SYS 및 SYSTEM과 같은 관리 계정의 암호를 지정합니다. 여기서는 모든 관리 계정에 대한 암호로 oracle\_4U를 사용합니다.

## DBCA를 사용하여 데이터베이스 생성



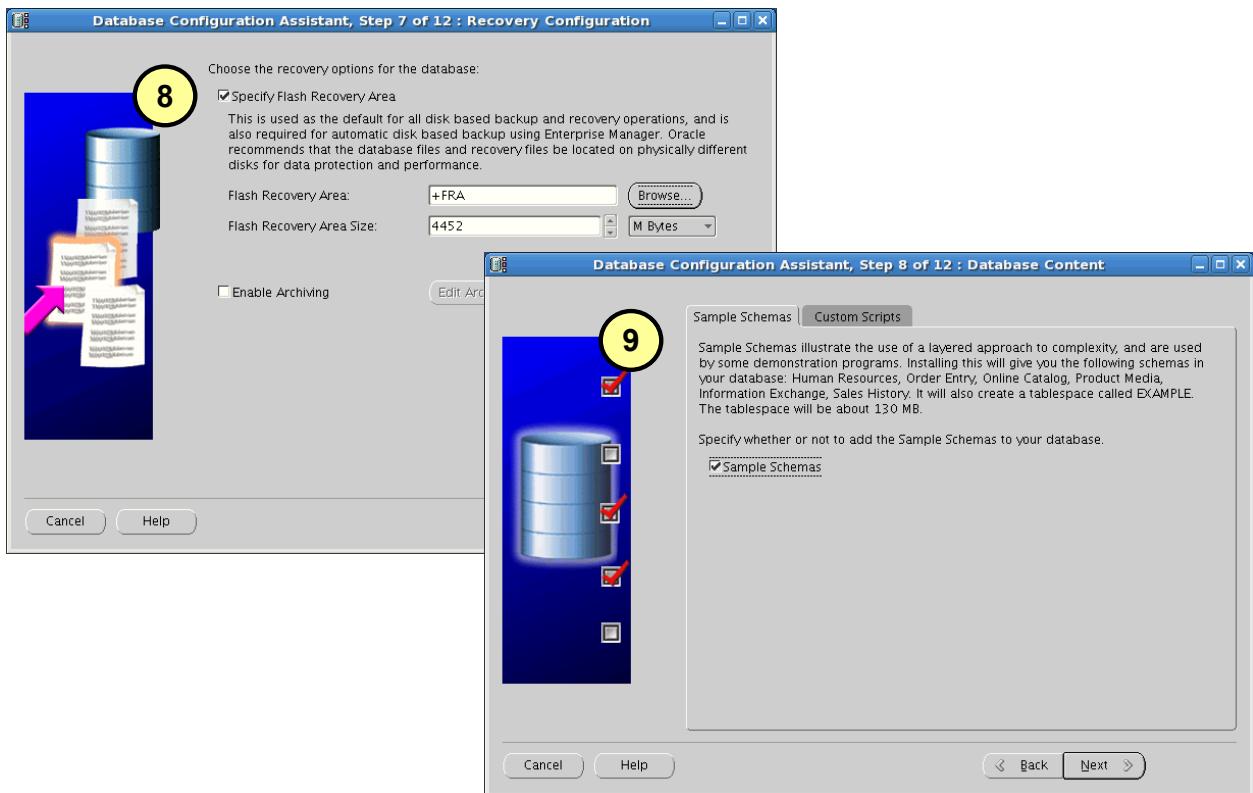
ORACLE

Copyright © 2010, Oracle. All rights reserved.

### DBCA를 사용하여 데이터베이스 생성(계속)

- 7. Database File Locations:** 데이터베이스에서 사용할 저장 영역 메커니즘 유형(자동 저장 영역 관리(ASM) 또는 파일 시스템)을 지정합니다. Storage Location에 대해 원하는 위치를 선택합니다. 이 예제에서는 OMF(Oracle 관리 파일)이 사용되므로 ASM이 저장 영역 메커니즘으로 사용됩니다. Browse 버튼을 사용하면 사용 가능한 디스크 그룹을 보고 가장 적합한 디스크 그룹(슬라이드 예제에서는 +DATA가 선택됨)을 선택할 수 있습니다. 디스크 그룹을 선택한 다음에는 ASM에 해당하는 ASMSNMP 암호를 제공하라는 메시지가 표시됩니다. OMF(Oracle 관리 파일)은 파일 시스템 저장 영역에서도 사용할 수 있으므로 오라클 데이터베이스에서 운영 체제 파일을 직접 관리할 필요가 없습니다. 필요한 경우 Multiplex Redo Logs 및 Control Files 버튼을 눌러 리두 로그 및 콘트롤 파일의 복사본을 추가로 생성할 수 있습니다. 다음과 같은 파일 위치 변수를 조정할 수 있는 버튼도 제공됩니다. ORACLE\_BASE, ORACLE\_HOME, DB\_NAME, DB\_UNIQUE\_NAME, SID.

# DBCA를 사용하여 데이터베이스 생성



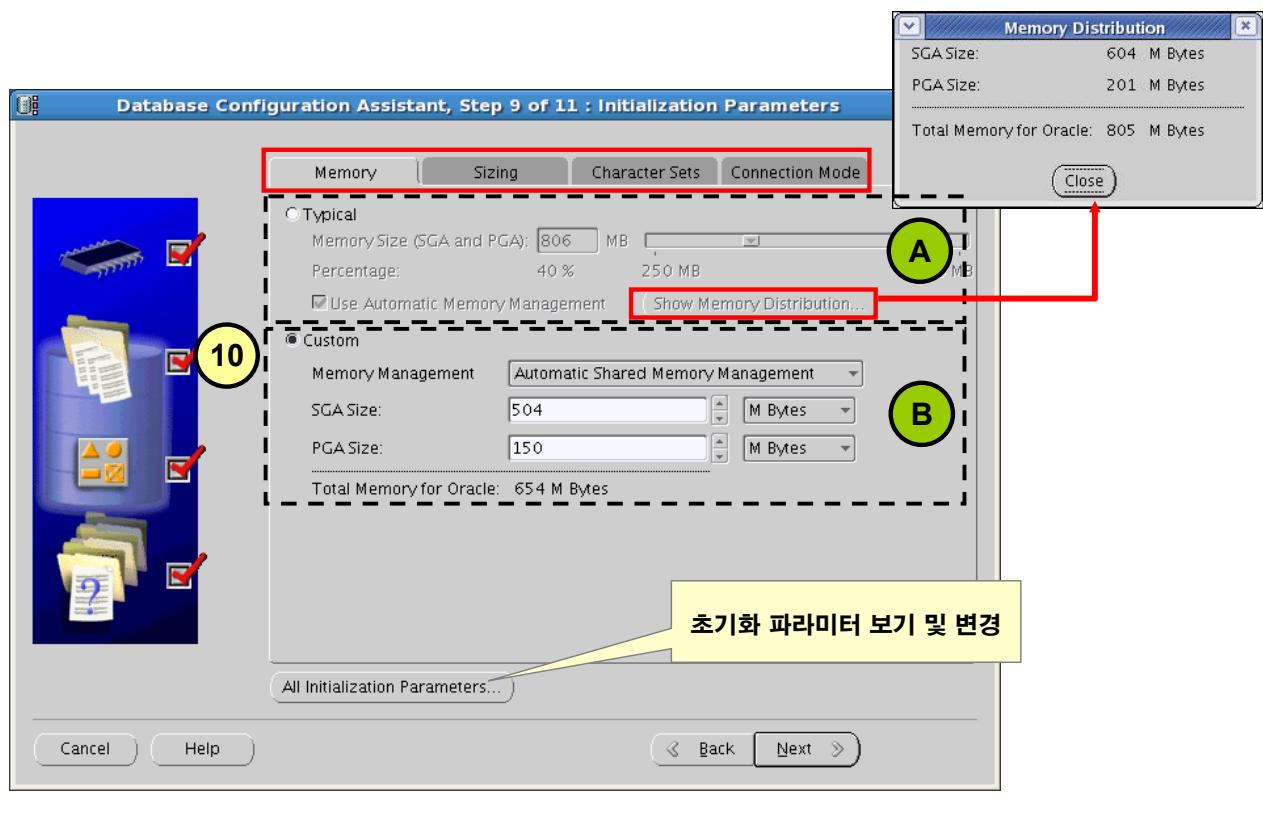
ORACLE

Copyright © 2010, Oracle. All rights reserved.

## DBCA를 사용하여 데이터베이스 생성(계속)

8. **Recovery Configuration:** 데이터베이스에 대한 recovery 옵션을 선택합니다. Fast Recovery Area를 구성하려면 Specify Flash Recovery Area 옵션을 선택한 다음 Fast Recovery Area의 위치와 크기를 지정합니다. Fast Recovery Area의 위치에 표시된 예제에서는 +FRA ASM 디스크 그룹을 사용하도록 설정되어 있으며 크기는 4452MB입니다. Fast Recovery Area의 크기는 데이터베이스 크기보다 큰 것이 좋습니다. 아카이브 및 관련 파라미터 아카이브도 이 화면에서 구성할 수 있습니다.  
**주:** Flash Recovery Area는 이름이 Fast Recovery Area로 바뀌었지만 현재까지는 일부 오라클 제품에서 Flash Recovery Area가 발견될 수 있습니다.
9. **Database Content:** 이 페이지에서는 데이터베이스가 생성된 다음 실행할 커스텀 스크립트를 지정할 수 있는 템과 구성 요소(예: Sample Schemas)를 선택하기 위한 옵션이 제공됩니다.

## DBCA를 사용하여 데이터베이스 생성



ORACLE

Copyright © 2010, Oracle. All rights reserved.

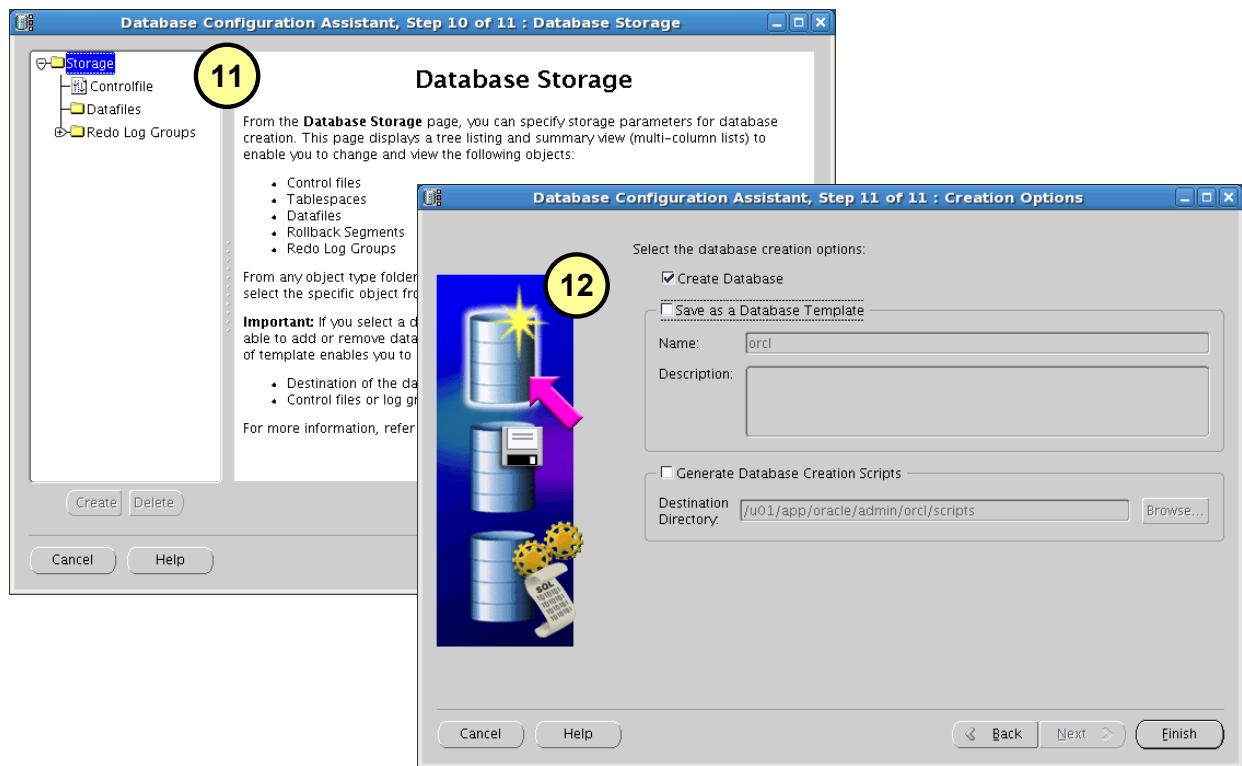
### DBCA를 사용하여 데이터베이스 생성(계속)

10. **Initialization Parameters:** 이 페이지의 템에서는 기본 데이터베이스 설정을 변경할 수 있는 페이지에 액세스할 수 있습니다.

- **Memory:** 이 페이지에서는 메모리 사용을 제어하는 초기화 파라미터를 설정합니다. (A) Typical 또는 (B) Custom 메모리 할당을 사용합니다.
- **Sizing:** 블록 크기를 지정하려면 바이트 단위로 크기를 입력하거나 기본값을 선택합니다.
- **Character Sets:** 이 페이지에서는 데이터베이스의 Character Set를 지정합니다.
- **최적의 사용법(Best Practice Tip):** 가능하면 다양한 언어 및 웹 기술을 유연하게 지원하는 유니코드를 데이터베이스 Character Set으로 사용할 것을 권장합니다.
- **Connection Mode:** Dedicated 또는 Shared Server 모드를 선택합니다. 자세한 사항은 "Oracle 네트워크 환경 구성" 단원을 참조하십시오.
- All Initialization Parameters 버튼을 눌러 초기화 파라미터를 보고 수정합니다.

**주:** DB\_BLOCK\_SIZE 파라미터와 같은 일부 초기화 파라미터는 한 번 설정하면 데이터베이스의 존속 기간 동안 계속 적용됩니다.

# DBCA를 사용하여 데이터베이스 생성



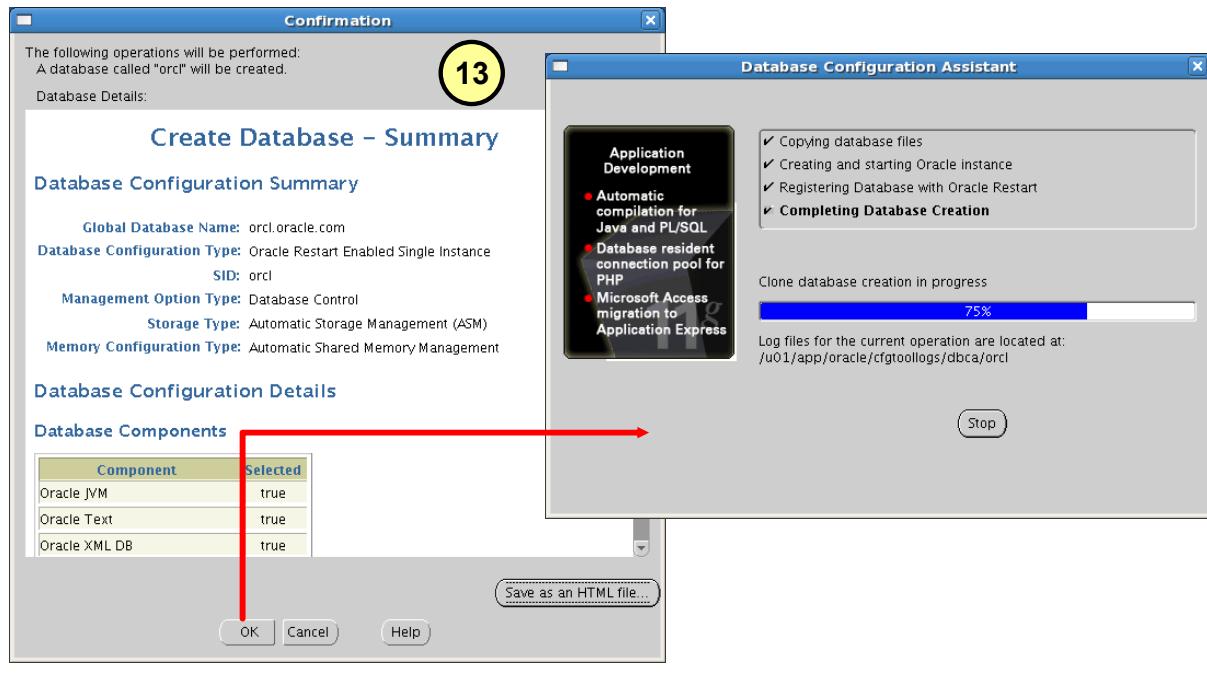
ORACLE

Copyright © 2010, Oracle. All rights reserved.

## DBCA를 사용하여 데이터베이스 생성(계속)

11. **Database Storage:** 필요한 경우 현재 데이터베이스 저장 영역 설정을 검토하고 수정합니다. 데이터베이스에 대해 미리 구성된 템플릿 중 하나를 선택한 경우 콘트롤 파일 또는 데이터 파일을 추가하거나 제거할 수 없습니다.
12. **Create Options:** 데이터베이스를 지금 생성하거나 데이터베이스 정의를 템플릿으로 저장하거나 스크립트를 생성하는 옵션이 제공됩니다. 모든 옵션을 선택하고 Finish 를 누르면 DBCA가 먼저 데이터베이스 템플릿을 저장하고 대상 디렉토리에 스크립트를 생성한 다음 마지막으로 데이터베이스를 생성합니다.

# 데이터베이스 요약 생성



**ORACLE**

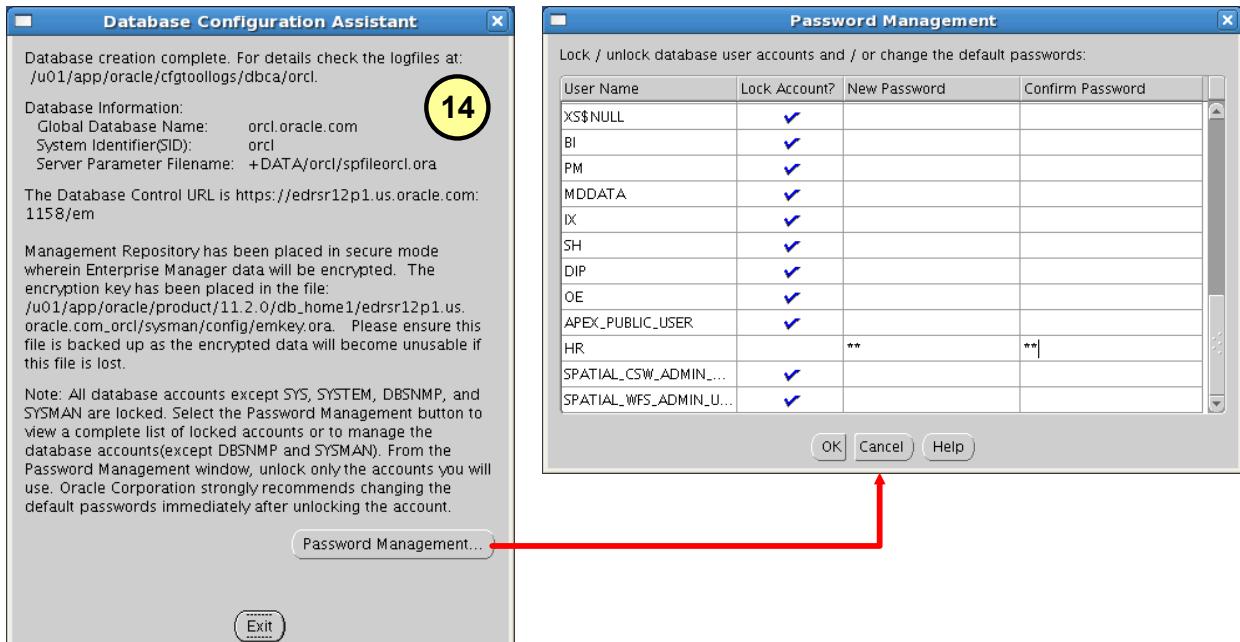
Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 요약 생성

- Finish를 누르면 생성 전 데이터베이스의 모든 구성은 검토할 수 있는 Confirmation 페이지가 표시됩니다. OK를 눌러 Confirmation 페이지를 닫고 데이터베이스 생성을 시작합니다.

**주:** 데이터베이스 정의를 HTML 파일로 저장하여 쉽게 참조할 수 있습니다.

## 암호 관리



ORACLE

Copyright © 2010, Oracle. All rights reserved.

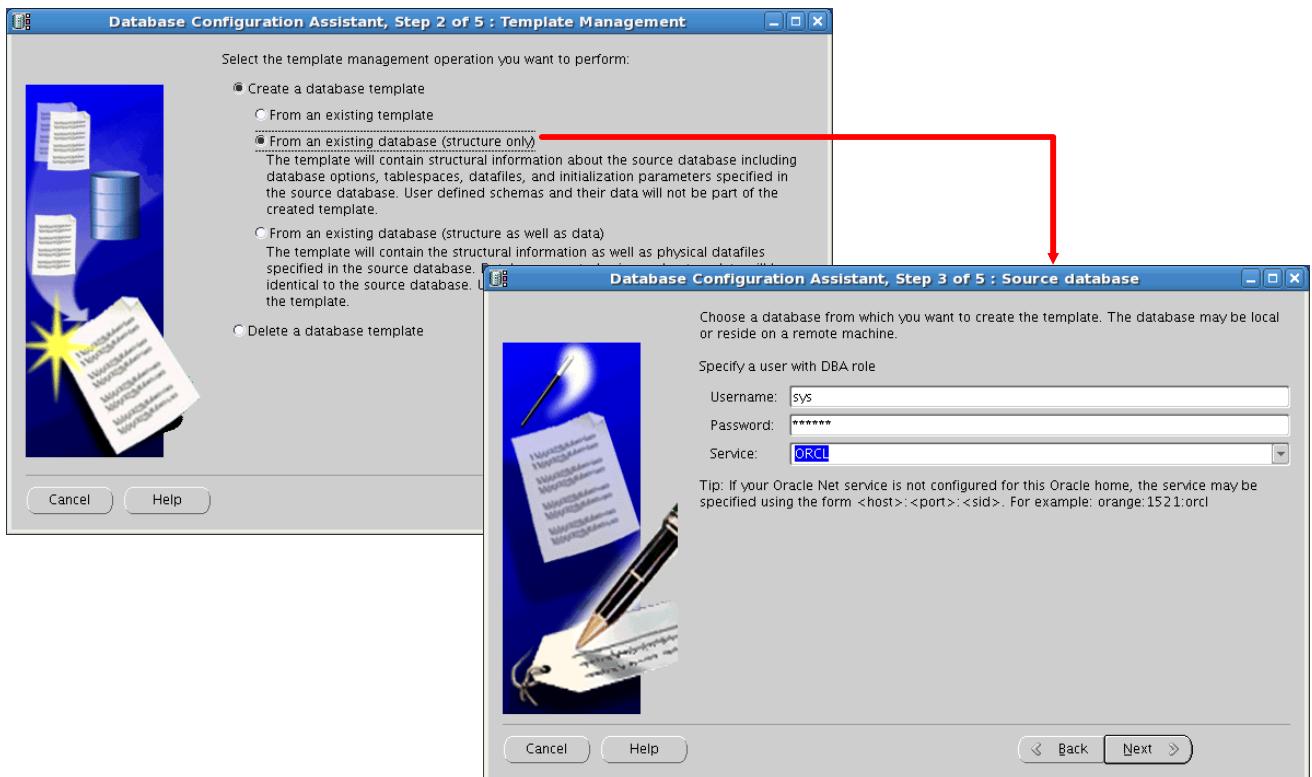
### 암호 관리

14. DBCA가 완료되면 나중에 참조할 수 있도록 다음 정보를 기록해 두십시오.

- 설치 로그 파일 위치
- Global Database Name
- SID(시스템 식별자)
- 서버 파라미터 파일 이름 및 위치
- Enterprise Manager URL

Password Management를 눌러 사용할 데이터베이스 계정의 Lock을 해제합니다. 계정의 Lock을 해제할 때 암호를 입력합니다. 지금 Lock을 해제하지 않은 계정은 이후에 필요에 따라 Lock을 해제할 수 있습니다.

## 데이터베이스 설계 템플리트 생성



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 설계 템플리트 생성

템플리트는 새 데이터베이스를 만드는 시작점으로 사용하는 미리 정의된 데이터베이스 정의입니다. 데이터베이스 생성 과정에서 템플리트를 생성하지 않은 경우 언제든지 DBCA 를 호출하고 Manage Templates 작업을 선택하여 생성할 수 있습니다.

다음과 같은 세 가지 방법으로 템플리트를 생성할 수 있습니다.

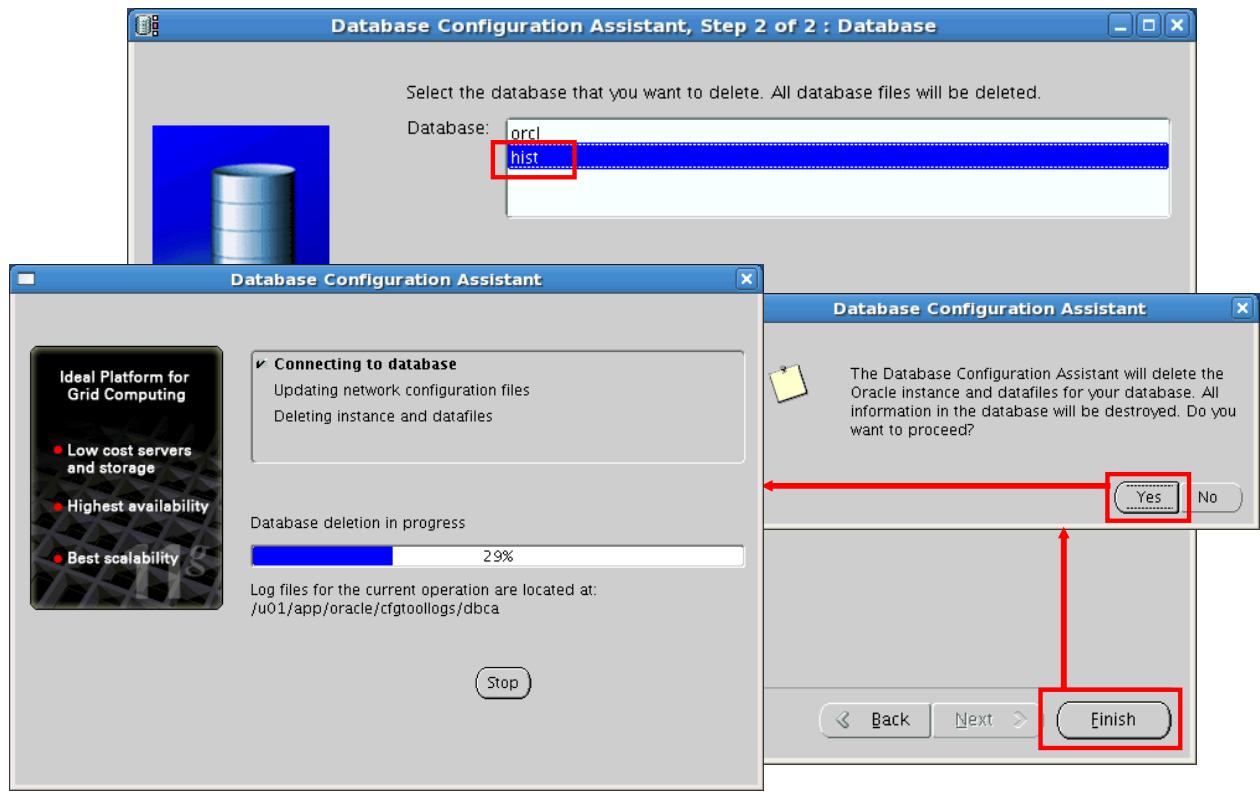
- 기존 템플리트로 생성
- 기존 데이터베이스로 생성(구조만 사용)
- 기존 데이터베이스로 생성(구조 및 데이터 사용)

DBCA는 데이터베이스 설계 템플리트 생성 단계를 안내합니다.

특정 템플리트가 더 이상 필요하지 않으면 DBCA의 Template Management 페이지에서 "Delete a database template" 옵션을 사용합니다.

**주:** 생성한 템플리트는 DBCA를 사용하여 새 데이터베이스를 생성할 때 Database Templates list 에 표시됩니다.

## DBCA를 사용하여 데이터베이스 삭제



### DBCA를 사용하여 데이터베이스 삭제

터미널 window에서 dbca를 입력하여 DBCA를 시작한 다음 Welcome 페이지에서 Next 를 누릅니다. 데이터베이스를 삭제하려면 다음 단계를 수행하십시오.

1. Operations 페이지에서 Delete a Database를 선택합니다. 그런 다음 Next를 누릅니다.
2. 삭제할 데이터베이스(여기서는 hist)를 선택하고 Finish를 누릅니다.
3. Yes를 눌러 삭제를 확인합니다.
4. 삭제가 완료되면 다른 작업을 수행할지 묻는 메시지가 표시됩니다. 적절한 옵션을 선택합니다.

**주:** 삭제하려는 데이터베이스는 DBCA가 파일 위치 정보를 확인하기 위해 데이터베이스에 연결할 수 있도록 작동되어 실행 중이어야 합니다.

## DBCA를 사용하여 데이터베이스 삭제(계속)

데이터베이스 삭제 과정에는 해당 데이터 파일, 리두 로그 파일, 콘트롤 파일 및 초기화 파라미터 파일을 제거하는 과정이 포함됩니다. DROP DATABASE SQL 문을 사용하여 데이터베이스를 수동으로 삭제할 수 있습니다. DROP DATABASE 문은 모든 콘트롤 파일과 콘트롤 파일에 나열된 다른 모든 데이터베이스 파일을 삭제합니다. DROP DATABASE 문을 성공적으로 사용하려면 다음 조건이 모두 충족되어야 합니다.

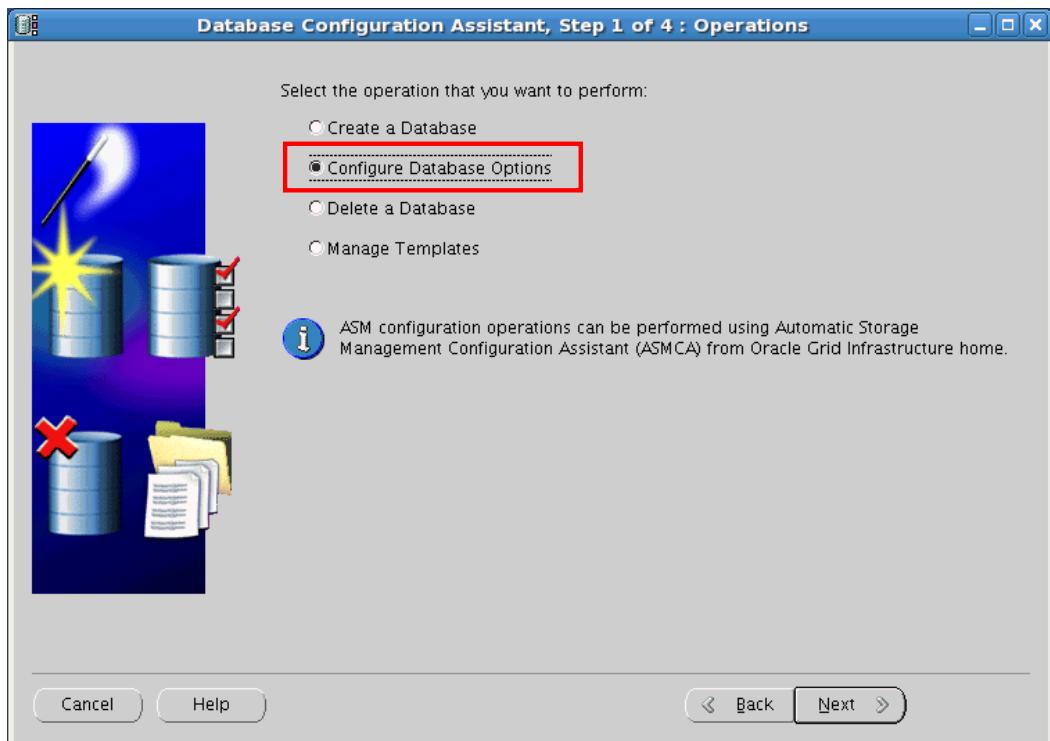
- 데이터베이스가 마운트되어 있고 닫혀 있어야 합니다.
- 데이터베이스가 공유 모드가 아니라 배타적으로 마운트되어 있어야 합니다.
- 데이터베이스가 RESTRICT 모드로 시작되어야 합니다.

이러한 명령문의 예는 다음과 같습니다.

```
STARTUP RESTRICT FORCE MOUNT;  
DROP DATABASE;
```

DROP DATABASE 문은 아카이브된 로그 파일 및 데이터베이스의 복사본이나 백업에 영향을 주지 않습니다. 이러한 파일을 삭제할 때는 RMAN(Recovery Manager)을 사용하는 것이 가장 좋습니다. 데이터베이스가 Raw Disk에 있는 경우 실제 Raw Disk 특수 파일은 삭제되지 않습니다.

## 추가 작업에 DBCA 사용



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 추가 작업에 DBCA 사용

DBCA를 사용하여 데이터베이스 옵션(예: Oracle Label Security 또는 Oracle Database Vault)을 구성할 수 있습니다. 데이터베이스 소프트웨어 설치 중에 모든 옵션이 기본적으로 설치되지는 않으므로 옵션을 사용하도록 데이터베이스를 구성하려고 시도하기 전에 이러한 옵션을 설치해야 합니다.

#### 주:

- Oracle Label Security에 대한 자세한 내용은 *Oracle Label Security Administrator's Guide*를 참조하십시오.
- Oracle Database Vault에 대한 자세한 내용은 *Oracle Database Vault Administrator's Guide*를 참조하십시오.

## 퀴즈

**DB\_BLOCK\_SIZE** 파라미터는 데이터베이스의 존속 기간 동안 사용되도록 설정되며 변경할 수 없습니다.

- 1. 참**
- 2. 거짓**

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

**해답: 1**

## 퀴즈

**DROP DATABASE 명령을 사용하여 데이터베이스를 삭제하려면  
데이터베이스가 다음과 같은 상태여야 합니다.**

- 1. 열려 있고 RESTRICT 모드여야 합니다.**
- 2. RESTRICT 모드에서 배타적으로 마운트되어야 합니다.**
- 3. Immediate 옵션으로 종료되어 있어야 합니다.**



Copyright © 2010, Oracle. All rights reserved.

**해답: 2**

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- DBCA(Database Configuration Assistant)를 사용하여 데이터베이스 생성
- DBCA를 사용하여 데이터베이스 생성 스크립트 작성
- DBCA를 사용하여 데이터베이스 설계 템플리트 관리
- DBCA를 사용하여 추가 작업 수행

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 3 개요: DBCA 사용

이 연습에서는 다음 내용을 다룹니다.

- DBCA를 사용하여 ORCL 데이터베이스 생성
- HR 스키마 Lock 해제

**주: 다음 모든 연습을 수행하려면 반드시 데이터베이스 생성을 완료하고 HR 스키마 Lock을 해제해야 합니다.**

- DBCA를 사용하여 ORCL 데이터베이스 설계 템플릿 생성
- DBCA를 사용하여 데이터베이스 생성 스크립트 작성

ORACLE®

Copyright © 2010, Oracle. All rights reserved.



# 데이터베이스 Instance 관리

4

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 오라클 데이터베이스와 구성 요소 시작 및 정지
- Oracle Enterprise Manager 사용
- SQL\*Plus를 사용하여 데이터베이스 액세스
- 데이터베이스 초기화 파라미터 수정
- 데이터베이스 시작 단계 설명
- 데이터베이스 종료 옵션 설명
- Alert Log 확인
- Dynamic Performance 뷰 액세스

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 관리 프레임워크

## Oracle Database 11g Release 2 관리 프레임워크

### 구성 요소:

- 데이터베이스 Instance
- 리스너
- 관리 인터페이스:
  - Database Control
  - 관리 에이전트(Grid Control 사용 시)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 관리 프레임워크

오라클 데이터베이스 관리 프레임워크에는 다음과 같은 세 가지 주요 구성 요소가 있습니다.

- 관리되고 있는 데이터베이스 instance
- 데이터베이스 연결을 허용하는 리스너
- 관리 인터페이스. 데이터베이스 서버가 실행되는 노드(해당 서버를 Oracle Enterprise Manager Grid Control에 연결)에서 실행되는 관리 에이전트 또는 독립형 Oracle Enterprise Manager Database Control이며 *Database Console*이라고도 합니다.

이러한 구성 요소 서비스를 사용하려면 각 구성 요소를 시작해야 하며 오라클 데이터베이스를 호스트하는 서버를 종료할 때 각 구성 요소를 정상적으로 종료해야 합니다.

# Database Control 시작 및 정지

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/db_home1
is /u01/app/oracle
$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release 11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation. All rights reserved.
http://host01.example.com:1158/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control .....started.

-----
Logs are generated in directory
/u01/app/oracle/product/11.2.0/db_home1/host01.example.com_orcl/sysman/
log
```

```
$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release 11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation. All rights reserved.
https://host01.example.com:1158/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.
```

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## Database Control 시작 및 정지

오라클 데이터베이스는 Grid Control 프레임워크에 연결되지 않은 데이터베이스에 대해 *Database Control*이라는 독립형 관리 콘솔을 제공합니다. Database Control로 관리되는 각 데이터베이스에는 별도의 Database Control이 설치되어 있으며 하나의 Database Control에서는 하나의 데이터베이스만 관리할 수 있습니다. Database Control을 사용하기 전에 dbconsole 프로세스가 시작되었는지 확인하십시오.

dbconsole 프로세스를 시작하려면 다음 명령을 실행하십시오.

```
emctl start dbconsole
```

dbconsole 프로세스를 정지하려면 다음 명령을 실행하십시오.

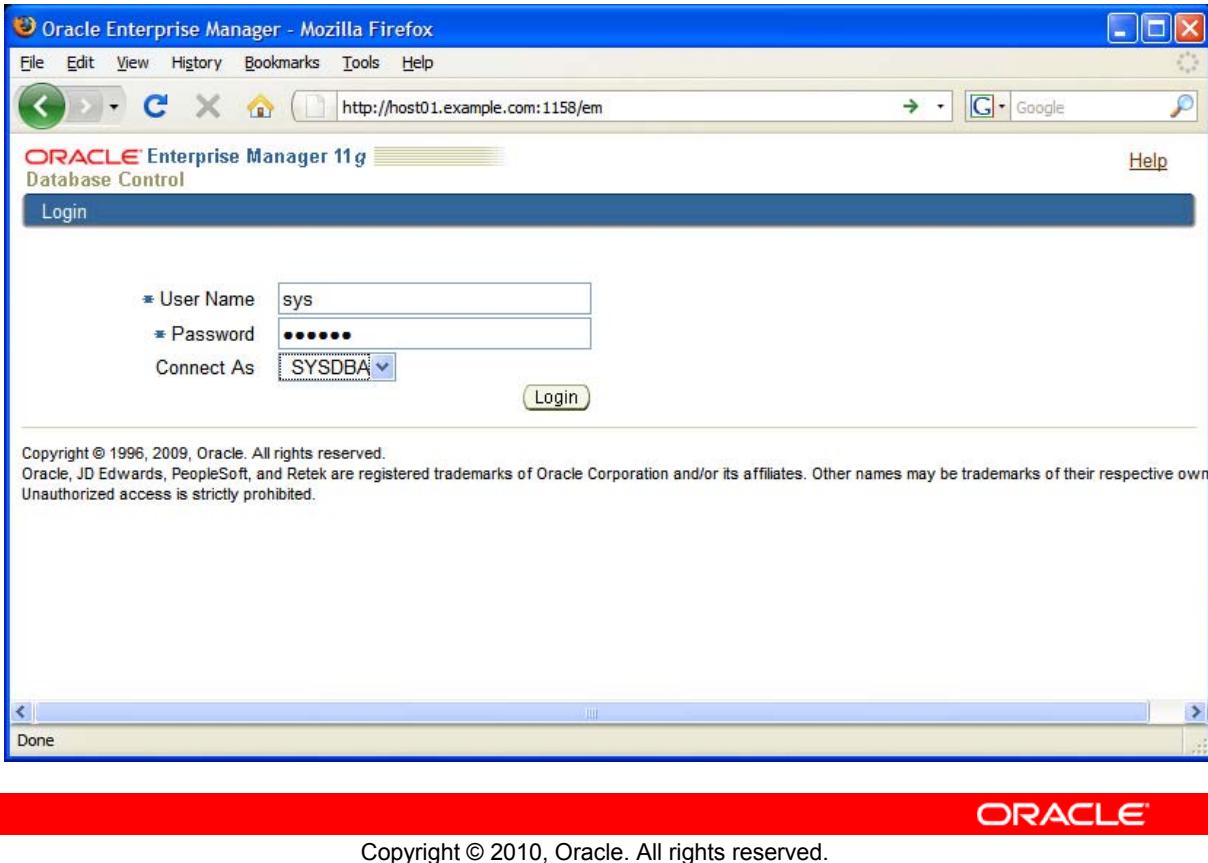
```
emctl stop dbconsole
```

dbconsole 프로세스의 상태를 확인하려면 다음 명령을 실행하십시오.

```
emctl status dbconsole
```

**주:** \$ORACLE\_HOME/bin 디렉토리가 OS(운영 체제) 경로에 없으면 해당 디렉토리로 이동해야 합니다. 그리드 Infrastructure가 설치된 경우 \$ORACLE\_HOME 위치가 두 개이며, 두 위치 모두 emctl 유ти리티를 포함합니다. emctl 유ти리티는 항상 오라클 데이터베이스의 \$ORACLE\_HOME을 사용하여 호출해야 하며, 그리드 Infrastructure의 \$ORACLE\_HOME을 사용해서는 안됩니다. Database Control에서는 서버측 에이전트 프로세스를 사용합니다. 이 에이전트 프로세스는 dbconsole 프로세스가 시작되거나 정지될 때 자동으로 시작 및 정지됩니다.

# Oracle Enterprise Manager



## Oracle Enterprise Manager

오라클 데이터베이스 소프트웨어를 설치하면 OUI(Oracle Universal Installer)가 Oracle Enterprise Manager(Enterprise Manager)도 함께 설치합니다. Enterprise Manager의 웹 기반 Database Control은 오라클 데이터베이스를 관리하는 기본 도구로 사용됩니다. Enterprise Manager는 DBA(데이터베이스 관리자)가 수행해야 하는 거의 모든 작업을 위한 그래픽 인터페이스를 제공합니다. Enterprise Manager를 사용하여 수행할 수 있는 작업에는 alert 요약 및 성능 그래프 보기, 객체 생성 및 수정, 백업 및 recovery 수행 등이 포함됩니다. 대부분의 경우에는 Enterprise Manager의 링크를 누르면 페이지의 내용을 보다 자세하게 확인할 수 있습니다.

**주:** Oracle Database 11g Release 2에서 Enterprise Manager에 액세스하는 URL은 보안 연결을 활성화하기 위해 HTTP가 아닌 HTTPS 프로토콜을 사용합니다. 그러므로 Enterprise Manager dbconsole에 액세스하려면 다음과 같은 형식으로 URL을 입력해야 합니다.

`https://machine_name:port/em`

시스템에서 생성하는 첫번째 데이터베이스의 경우 Enterprise Manager Database Control에 액세스하기 위한 기본 포트 번호는 1158입니다. 이 번호는 다른 번호로 변경할 수도 있으며, 특히 동일 호스트에 데이터베이스가 여러 개 있는 경우에 번호를 변경할 수 있습니다. 포트 번호는 portlist.ini 파일에서 확인할 수 있습니다. 일부 오라클 데이터베이스 응용 프로그램에 대한 포트는 \$ORACLE\_HOME/install 디렉토리에 있는 portlist.ini 파일에 나열되어 있습니다.

## Oracle Enterprise Manager(계속)

Enterprise Manager에 대한 URL을 입력하는 경우 표시되는 컨텐트는 데이터베이스의 상태에 따라 달라집니다.

- 데이터베이스가 작동 중이면 Enterprise Manager에 Database Control Login 페이지가 표시됩니다. Database Control에 대한 액세스 권한이 있는 username을 사용하여 데이터베이스에 로그인합니다. 처음에는 SYS, SYSMAN 또는 SYSTEM이 이러한 유저에 해당됩니다. 데이터베이스를 설치할 때 해당 계정에 지정한 암호를 사용합니다. Connect As 옵션에서 Normal 또는 SYSDBA를 선택하여 특수 데이터베이스 관리 권한으로 데이터베이스에 로그인합니다.
- 데이터베이스가 작동 중지된 상태인 경우에는 Enterprise Manager에 "Startup/Shutdown and Perform Recovery" 페이지가 표시됩니다. 이런 경우에는 Startup/Shutdown 버튼을 누릅니다. 그러면 입력해야 하는 호스트 및 대상 데이터베이스 로그인 username 및 암호를 입력하라는 메시지가 표시됩니다.

**주:** Enterprise Manager를 시작하는 데 문제가 있으면 리스너가 시작되었는지 확인하십시오.

# Database Home 페이지

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. At the top, it says "ORACLE Enterprise Manager 11g Database Control". The top navigation bar includes "Setup", "Preferences", "Help", "Logout", and "Database". It also shows "Logged in As SYS". Below the header, it says "Database Instance: orcl.example.com". The main menu has tabs: Home, Performance, Availability, Server, Schema, Data Movement, Software and Support, and the highlighted "속성 페이지". A status message indicates the page was refreshed on Jun 18, 2009 at 11:46:00 PM GMT+07:00. The "View Data" dropdown is set to "Automatically (60 sec)".

**General**

- Status: Up (green arrow icon)
- Up Since: Jun 18, 2009 5:31:03 AM GMT+07:00
- Instance Name: orcl
- Version: 11.2.0.1.0
- Host: edrsr25p1.us.oracle.com
- Listener: LISTENER\_edrsr25p1.us.oracle.com
- ASM: +ASM\_edrsr25p1.us.oracle.com

[View All Properties](#)

**Host CPU**

**Active Sessions**

**SQL Response Time**

**Diagnostic Summary**

ADDM Findings	2
Period Start Time	Jun 18, 2009 10:00:40 PM GMT+07:00
Alert Log	No ORA- errors
Active Incidents	0

**Space Summary**

Database Size (GB)	1.448
Problem Tablespaces	0
Segment Advisor Recommendations	0
Policy Violations	0
Dump Area Used (%)	85

**High Availability**

Console Oracle Restart	Enabled
Instance Recovery Time (sec)	14
Last Backup	n/a
Usable Flash Recovery Area (%)	95.96
Flashback Database Logging	Disabled

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## Database Home 페이지

Database Home 페이지는 데이터베이스의 전반적인 상태를 나타내는 여러 metric을 통해 데이터베이스의 현재 상태를 보여줍니다. 템(tab)이라고도 하는 속성 페이지를 사용하여 데이터베이스 관리를 위한 Performance, Availability, Server, Schema, Data Movement 및 Software and Support 페이지에 액세스할 수 있습니다.

Database Home 페이지에서는 다음과 같은 데이터베이스 instance의 성능 및 상태 정보를 확인할 수 있습니다.

- Instance 이름, 데이터베이스 버전, Oracle 홈 위치, media recovery 옵션 및 기타 관련된 instance 데이터
- 현재 instance 가용성
- 미해결된 alert
- 세션 및 SQL 관련 성능 정보
- 주요 공간 사용량 metrics
- 더 자세한 정보를 제공하는 드릴 다운 링크(예: LISTENER\_<host\_name>)

## 다른 오라클 도구

구성 요소  
 > SQL\*Plus  
 초기화 파라미터  
 DB 시작  
 DB 종료  
 Alert Log  
 성능 뷰

- **SQL\*Plus에서는 다음을 수행할 수 있도록 데이터베이스에 대해 추가 인터페이스를 제공합니다.**
  - 데이터베이스 관리 작업 수행
  - 데이터베이스에서 데이터를 query, 삽입, 갱신 및 삭제하는 SQL 명령 실행
- **SQL Developer:**
  - 오라클 데이터베이스 Instance에 액세스하기 위한 Graphical User Interface
  - SQL 및 PL/SQL을 사용한 개발을 모두 지원
  - 기본 오라클 데이터베이스 설치에서 사용 가능

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 다른 오라클 도구

Enterprise Manager 외에도 SQL\*Plus 및 SQL Developer를 사용하여 SQL 문을 실행할 수 있습니다. 이러한 도구를 사용하여 데이터베이스에서 데이터 선택, 삽입, 갱신 또는 삭제를 비롯한 다양한 데이터베이스 관리 작업을 수행할 수 있습니다.

SQL\*Plus는 SQL 및 PL/SQL 문을 오라클 데이터베이스에 제출(submit)하는 데 사용하는 명령행 프로그램입니다. 명령문은 대화형으로 제출(submit)할 수도 있고 SQL\*Plus 스크립트로 제출(submit)할 수도 있습니다. SQL\*Plus는 데이터베이스와 함께 설치되며 \$ORACLE\_HOME/bin 디렉토리에 있습니다.

SQL\*Plus는 명령행에서 또는 Windows 클라이언트의 시작 메뉴에서 시작할 수 있습니다.

SQL Developer는 오라클 데이터베이스 Instance에 액세스하기 위한 Graphical User Interface로, SQL 및 PL/SQL 언어를 사용한 개발을 모두 지원하며 기본 오라클 데이터베이스 설치에서 사용할 수 있습니다.

SQL Developer를 사용하면 데이터베이스 객체를 찾아보고, SQL 문 및 SQL 스크립트를 실행하고, PL/SQL 문을 편집 및 디버그할 수 있습니다. 또한 제공된 보고서를 원하는 수만큼 실행할 수도 있고 보고서를 직접 생성하여 저장할 수도 있습니다.

**주:** 이 과정에서는 Enterprise Manager 및 SQL\*Plus를 사용합니다.

# SQL\*Plus 사용

**SQL\*Plus의 특징은 다음과 같습니다.**

- 명령행 도구
- 대화식 또는 일괄 처리 모드에서 사용

```
$ sqlplus hr

SQL*Plus: Release 11.2.0.1.0 - Production on Thu Jun 18 05:04:49 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Enter Password: *****

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

SQL> select last_name from employees;
LAST_NAME
-----
Abel
Ande
...
```

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## SQL\*Plus 사용

SQL\*Plus의 명령행 인터페이스를 통해 SQL\*Plus, SQL 및 PL/SQL 명령을 실행하여 다음 작업을 수행할 수 있습니다.

- SQL 명령 및 PL/SQL 블록 입력, 편집, 실행, 검색 및 저장
- query 결과 형식 지정, 계산, 저장 및 인쇄
- 테이블에 대한 열 정의 나열
- 일반 유저에게 메시지를 보내고 응답 받기
- 데이터베이스 관리 수행

SQL\*Plus를 시작하려면 다음을 수행하십시오.

1. 터미널 window를 엽니다.
2. 명령행 프롬프트에 SQL\*Plus 명령을 다음 형식으로 입력합니다.

```
$ sqlplus <userid>/<pwd> or /nolog
```

3. NOLOG 옵션을 사용하는 경우에는 CONNECT 뒤에 연결에 사용할 Username을 붙여 입력해야 합니다.

```
SQL> connect <username>
```

4. 암호 입력 화면이 나타나면 유저의 암호를 입력합니다. SQL\*Plus가 시작되고 기본 데이터베이스에 연결됩니다.

## 셀 스크립트에서 SQL\*Plus 호출

```
$ ./batch_sqlplus.sh
SQL*Plus: Release 11.2.0.1.0 - Production on Thu Jun 18 05:10:19 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

SQL>
COUNT(*)
-----
107
SQL>
107 rows updated.
SQL>
Commit complete.

# Name of this file: batch_sqlplus.sh
# Count employees and give raise.
sqlplus hr/hr <<EOF
select count(*) from employees;
update employees set salary = salary*1.10;
commit;
quit
EOF

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
$
```

출처

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 셀 스크립트에서 SQL\*Plus 호출

sqlplus를 호출하고 운영 체제의 파라미터 전달 스크립트 구문을 사용하여 셀 스크립트나 BAT 파일에서 SQL\*Plus를 호출할 수 있습니다.

이 예제에서는 SELECT, UPDATE 및 COMMIT 문이 실행된 후 SQL\*Plus에서 운영 체제로 콘트롤이 반환됩니다.

## SQL\*Plus에서 SQL 스크립트 호출

script.sql

```
select * from departments where location_id = 1400;
quit
```

출력

```
$ sqlplus hr/hr @script.sql
```

```
SQL*Plus: Release 11.2.0.1.0 - Production on Thu Jun 18 05:13:42 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

Connected to:

```
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
60	IT	103	1400

```
Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### SQL\*Plus에서 SQL 스크립트 호출

SQL\*Plus 내부에서 기존 SQL 스크립트 파일을 호출할 수 있습니다. 슬라이드에 있는 것처럼 처음 SQL\*Plus를 호출할 때 명령행에서 파일을 호출할 수 있습니다. SQL\*Plus 세션 내에서 "@" 연산자를 사용하여 해당 파일을 호출할 수도 있습니다. 예를 들어, 다음 명령은 이미 설정된 SQL\*Plus 세션 내에서 스크립트를 실행합니다.

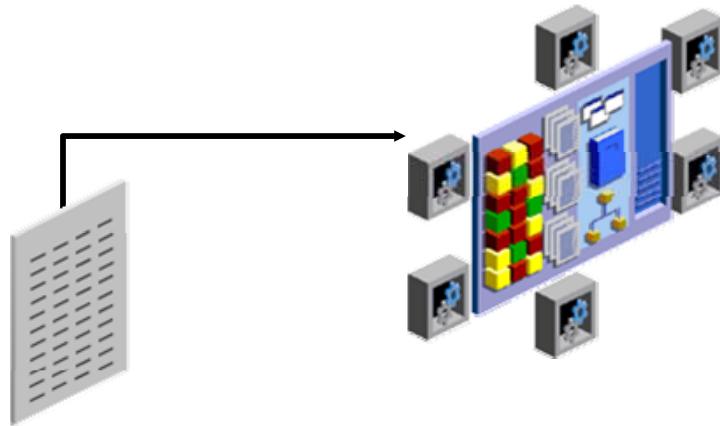
```
SQL> @script.sql
```

**주:** 스크립트 파일의 기본 파일 확장자는 .sql입니다. 저장 명령을 사용하여 SQL\*Plus에서 스크립트를 저장하면 확장자가 자동으로 적용됩니다. 이 확장자가 붙은 스크립트는 다음 예제와 같이 실행 시에 확장자를 제공하지 않아도 실행됩니다.

```
SQL> @script
```

# 초기화 파라미터 파일

구성 요소  
SQL\*Plus  
> 초기화 파라미터  
DB 시작  
DB 종료  
Alert Log  
성능 뷰



**spfileorcl.ora**

또는

**initorcl.ora**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 초기화 파라미터 파일

instance가 시작되면 초기화 파라미터 파일이 읽혀집니다. 다음과 같은 두 가지 유형의 파라미터 파일이 있습니다.

- **서버 파라미터 파일(SPFILE):** 일반적으로 사용되는 초기화 파라미터 파일 유형입니다. 데이터베이스 서버가 읽고 쓸 수 있는 이진 파일이며 수동으로 편집해서는 안됩니다. 이 파일은 Oracle Instance가 실행되는 서버에 있으며 종료 및 시작과 관계없이 계속 유지됩니다. 이 파일의 기본 이름은 spfile<SID>.ora이며 시작 시 자동으로 검색됩니다.
- **텍스트 초기화 파라미터 파일:** 데이터베이스 서버는 이러한 유형의 초기화 파라미터 파일을 읽을 수는 있지만 쓸 수는 없습니다. 종료 및 시작에 관계없이 초기화 파라미터 설정을 계속 유지하려면 텍스트 편집기를 사용하여 해당 설정을 수동으로 설정하고 변경해야 합니다. 이 파일의 기본 이름은 init<SID>.ora이며 SPFILE을 찾을 수 없는 경우 시작 시 자동으로 검색됩니다.

SPFILE을 생성하여 초기화 파라미터를 동적으로 유지 관리할 것을 권장합니다.

주: 오라클 데이터베이스는 Linux의 \$ORACLE\_HOME/dbs 디렉토리에서 초기화 파일을 검색합니다. ASM에서는 SPFILE이 대부분 ASM 디스크 그룹에 있습니다. 이 경우 init<SID>.ora 파일은 SPFILE 위치를 식별하는 \$ORACLE\_HOME/dbs 디렉토리에 있습니다.

## 초기화 파라미터 파일(계속)

### 초기화 파라미터의 값 유형

오라클 데이터베이스 서버에는 다음과 같은 유형의 초기화 파라미터 값이 있습니다.

- Boolean
- String
- Integer
- 파라미터 파일
- Reserved
- Big Integer

### 파생된 파라미터 값

일부 초기화 파라미터는 파생됩니다(즉, 해당 값이 다른 파라미터 값으로부터 계산됨).

일반적으로 파생 파라미터의 값은 변경해서는 안됩니다. 그러나 파생 파라미터의 값을 변경하는 경우 지정하는 값이 계산된 값을 재정의합니다.

예를 들어, SESSIONS 파라미터의 기본값은 PROCESSES 파라미터의 값에서 파생됩니다. PROCESSES의 값이 변경되면 지정된 값으로 재정의하지 않는 한 SESSIONS의 기본값도 변경됩니다.

### 운영 체제에 종속적인 파라미터 값

일부 초기화 파라미터의 유효한 값이나 값 범위는 호스트 운영 체제에 따라 달라집니다.

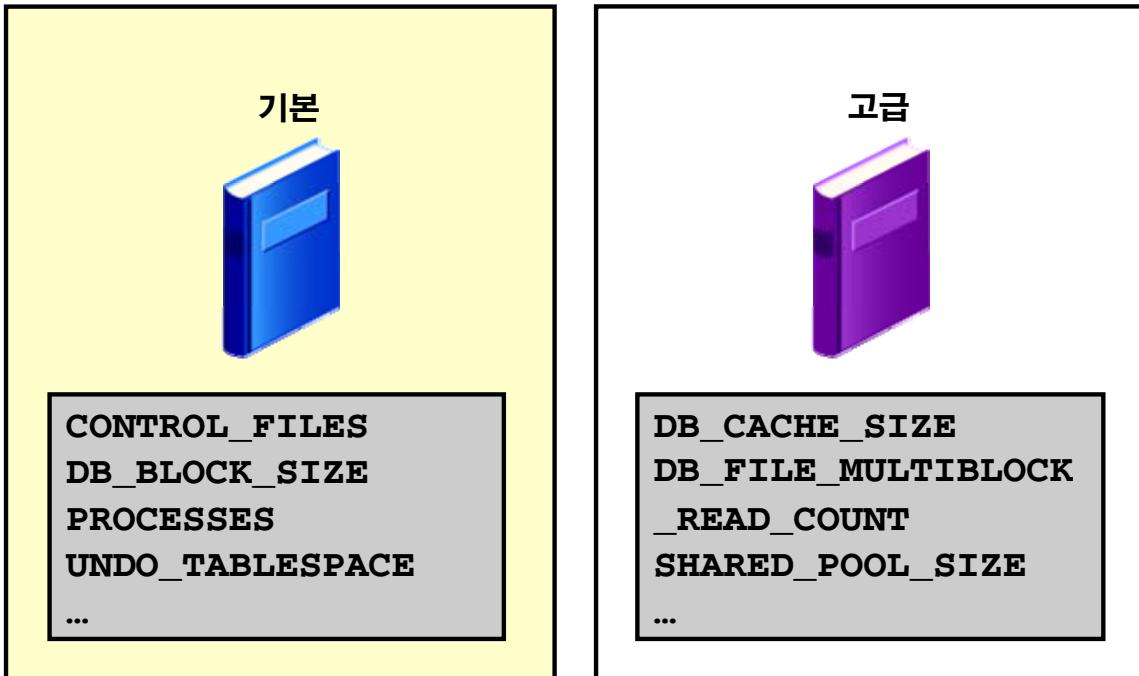
예를 들어, DB\_FILE\_MULTIBLOCK\_READ\_COUNT 파라미터는 순차 스캔 중에 단일 I/O 작업에서 읽는 최대 블록 수를 지정하며, 플랫폼에 종속됩니다. DB\_BLOCK\_SIZE에 의해 설정되는 해당 블록 크기의 기본값은 운영 체제에 따라 달라집니다.

### 파라미터 값 설정

초기화 파라미터를 활용하면 시스템 성능을 크게 개선할 수 있습니다. 일부 파라미터의 경우 용량 한계를 설정하지만 성능에는 영향을 주지 않습니다. 예를 들어, OPEN\_CURSORS의 값이 10이면 11번째 커서를 열려는 User Process에서는 오류를 받게 됩니다. 또한 성능에는 영향을 주지만 절대 한계는 적용하지 않는 파라미터도 있습니다. 예를 들어, OPEN\_CURSORS의 값을 줄여도 성능은 느려질 수 있지만 작업을 계속 수행할 수 있습니다.

파라미터 값을 늘리면 시스템 성능이 개선될 수 있지만 대부분의 파라미터를 늘리면 SGA(System Global Area) 크기도 늘어납니다. SGA의 크기가 늘어나면 데이터베이스 성능이 일정 수준 향상됩니다. 그러나 가상 메모리 운영 체제에서 SGA가 너무 커서 메모리에서 스왑인 및 스왑아웃이 수행되면 성능이 저하될 수 있습니다. 가상 메모리 작업 영역을 제어하는 운영 체제 파라미터는 SGA 크기를 염두에 두고 설정해야 합니다. 운영 체제 구성에 의해 SGA의 최대 크기에 한계가 적용될 수도 있습니다.

# 간소화된 초기화 파라미터



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 간소화된 초기화 파라미터

초기화 파라미터는 기본 및 고급의 두 가지 유형으로 제공됩니다.

대부분의 경우 30개의 기본 파라미터만 설정 및 튜닝하면 적절한 성능으로 데이터베이스를 운영할 수 있습니다. 하지만 고급 파라미터를 수정하여 성능을 최적화해야 하는 경우도 가끔 발생할 수 있습니다. 고급 파라미터는 약 314개가 있습니다.

기본 파라미터는 데이터베이스를 최적의 성능으로 계속 실행하기 위해 설정하는 파라미터입니다. 이 파라미터 이외의 모든 파라미터는 고급 파라미터라고 할 수 있습니다.

다음은 기본 파라미터의 예제입니다.

- Global Database Name: DB\_NAME 및 DB\_DOMAIN
- Fast Recovery Area 및 크기 지정: DB\_RECOVERY\_FILE\_DEST 및 DB\_RECOVERY\_FILE\_DEST\_SIZE
- 모든 SGA 구성 요소의 전체 크기 지정: SGA\_TARGET
- Undo space 관리 테이블스페이스 방법 지정: UNDO\_TABLESPACE
- COMPATIBLE 초기화 파라미터 및 취소 불가능한 호환성

**주:** 다음 페이지에는 일부 초기화 파라미터가 나와 있습니다. 전체 리스트를 보려면 *Oracle Database Reference*를 참조하십시오.

## 초기화 파라미터: 예제

파라미터	지정 내용
CONTROL_FILES	하나 이상의 콘트롤 파일 이름
DB_FILES	최대 데이터베이스 파일 수
PROCESSES	동시에 연결할 수 있는 최대 OS User Process 수
DB_BLOCK_SIZE	모든 테이블스페이스에서 사용하는 표준 데이터베이스 블록 크기
DB_CACHE_SIZE	표준 블록 버퍼 캐시 크기

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 초기화 파라미터: 예제

**CONTROL\_FILES** 파라미터: 하나 이상의 콘트롤 파일 이름을 지정합니다. 콘트롤 파일을 다중화하고 mirroring할 것을 강력히 권장합니다. 값 범위는 1~8개의 파일 이름(경로 이름 포함)이고 기본값은 운영 체제에 따라 다릅니다.

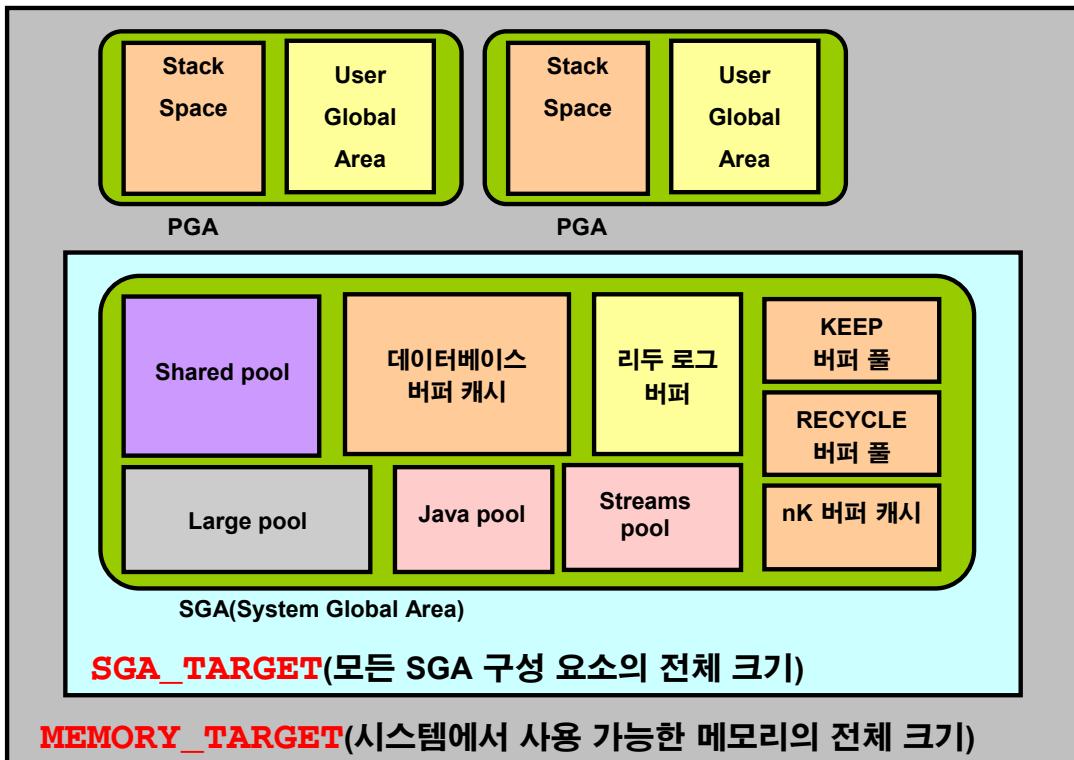
**DB\_FILES** 파라미터: 해당 데이터베이스에 대해 열 수 있는 최대 데이터베이스 파일 수를 지정합니다. 값 범위는 OS에 따라 다르고, 기본값은 200입니다.

**PROCESSES** 파라미터: Oracle 서버에 동시에 연결할 수 있는 OS User Process의 최대 수를 지정합니다. 이 값은 모든 백그라운드 프로세스 및 User Precess에 대해 허용되어야 합니다. 최소값은 6이고 최대값은 운영 체제에 따라 다르며, 기본값은 100입니다.

**DB\_BLOCK\_SIZE** 파라미터: 오라클 데이터베이스 블록의 크기를 바이트 단위로 지정합니다. 이 값은 데이터베이스를 생성할 때 설정되며 차후에 변경할 수 없습니다. 이 파라미터는 데이터베이스의 표준 블록 크기를 지정합니다. 모든 테이블스페이스는 기본적으로 이 크기를 사용합니다. 값 범위는 2048~32768(운영 체제에 따라 다름)이며, 기본값은 8192입니다.

**DB\_CACHE\_SIZE** 파라미터: 표준 블록 버퍼 캐시의 크기를 지정합니다. 값 범위는 16MB 이상이고 기본값은 SGA\_TARGET이 설정된 경우 0, 그렇지 않으면 48MB 또는 4MB\*cpu\_count 중 큰 값입니다.

## 초기화 파라미터: 예제



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 초기화 파라미터: 예제(계속)

SGA\_TARGET은 모든 SGA 구성 요소의 전체 크기를 지정합니다. SGA\_TARGET을 지정하는 경우 다음 메모리 풀의 크기가 자동으로 지정됩니다.

- 버퍼 캐시(DB\_CACHE\_SIZE)
- Shared pool(SHARED\_POOL\_SIZE)
- Large pool(LARGE\_POOL\_SIZE)
- Java pool(JAVA\_POOL\_SIZE)
- Streams pool(STREAMS\_POOL\_SIZE)

자동으로 튜닝되는 이러한 메모리 풀이 0이 아닌 값으로 설정되면 해당 값은 ASMM(자동 공유 메모리 관리)에서 최소 레벨로 사용됩니다. 응용 프로그램 구성 요소가 제대로 작동하는 데 최소한의 메모리가 필요한 경우 최소값을 설정합니다.

다음 풀은 크기가 수동으로 지정되는 구성 요소로 ASMM의 영향을 받지 않습니다.

- 로그 버퍼
- KEEP 및 RECYCLE 등의 기타 버퍼 캐시 및 기타 블록 크기
- 고정 SGA 및 기타 내부 할당

이러한 풀에 할당된 메모리는 ASMM이 활성화된 경우 SGA\_TARGET에서 사용할 수 있는 전체 메모리에서 감산됩니다.

**주:** MMON 프로세스는 ASMM을 지원하기 위한 자동 튜닝된 메모리 풀의 값을 계산합니다.

## 초기화 파라미터: 예제(계속)

MEMORY\_TARGET은 Oracle 시스템 전체에서 사용 가능한 메모리를 지정합니다. 데이터베이스는 필요에 따라 SGA 및 PGA를 늘리거나 줄이면서 MEMORY\_TARGET 값에 맞게 메모리를 튜닝합니다.

텍스트 기반 초기화 파라미터 파일에서는 MEMORY\_MAX\_TARGET을 생략하고 MEMORY\_TARGET 값을 포함한 경우 데이터베이스가 자동으로 MEMORY\_MAX\_TARGET 을 MEMORY\_TARGET 값으로 설정합니다. MEMORY\_TARGET의 행을 생략하고 MEMORY\_MAX\_TARGET에 대한 값을 포함하면 MEMORY\_TARGET 파라미터가 기본적으로 0이 됩니다. 시작한 후에는 MEMORY\_TARGET이 MEMORY\_MAX\_TARGET의 값을 초과하지 않는 경우 0이 아닌 값으로 동적으로 변경할 수 있습니다. MEMORY\_TARGET 파라미터는 ALTER SYSTEM 명령을 사용하여 수정할 수 있습니다. 값의 범위는 152MB~MEMORY\_MAX\_TARGET입니다.

## 초기화 파라미터: 예제

파라미터	지정 내용
PGA_AGGREGATE_TARGET	모든 서버 프로세스에 할당되는 PGA 메모리의 양
SHARED_POOL_SIZE	Shared Pool의 크기(바이트 단위)
UNDO_MANAGEMENT	사용할 Undo space 관리 모드

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 초기화 파라미터: 예제(계속)

**PGA\_AGGREGATE\_TARGET** 파라미터: Instance에 연결된 모든 서버 프로세스에 할당되는 PGA(Program Global Area) 메모리의 양을 지정합니다. 이 메모리는 SGA(System Global Area)에 포함되지 않습니다. 데이터베이스는 사용할 PGA 메모리의 대상 크기로 이 파라미터를 사용합니다. 이 파라미터를 설정하려면 Oracle Instance에서 사용할 수 있는 시스템의 전체 메모리에서 SGA를 뺍니다. 값의 범위는 정수와 단위를 지정하는 문자 K, M 또는 G(각각 킬로바이트, 메가바이트, 기가바이트)로 구성됩니다. 최소값은 10MB이고 최대값은 (4096 GB - 1)이며, 기본값은 10MB와 SGA 크기의 20% 중 더 큰 값입니다.

**SHARED\_POOL\_SIZE** 파라미터: Shared Pool의 크기를 바이트 단위로 지정합니다. Shared Pool에는 공유 커서, 내장 프로시저, 제어 구조 및 병렬 실행 메시지 버퍼 등과 같은 객체가 포함됩니다. 값 범위는 OS에 따라 다르고, 기본값은 SGA\_TARGET이 설정된 경우 0, 그렇지 않으면 64비트의 경우 128MB, 32비트의 경우 48MB입니다.

**UNDO\_MANAGEMENT** 파라미터: 시스템에서 사용할 undo space 관리 모드를 지정합니다. 이 파라미터를 AUTO로 설정하면 Instance는 AUM(Automatic Undo Management) 모드에서 시작되고. 그렇지 않은 경우 instance는 RBU(Rollback Undo) 모드로 시작됩니다. RBU 모드에서 undo space는 롤백 세그먼트로 할당됩니다. AUM 모드에서 undo space는 언두 테이블스페이스로 할당됩니다. 값 범위: AUTO 또는 MANUAL. 첫번째 Instance가 시작될 때 UNDO\_MANAGEMENT 파라미터를 생략하면 기본값인 AUTO가 사용됩니다.

## SQL\*Plus를 사용하여 파라미터 확인

```

SQL> SELECT name , value FROM V$PARAMETER;
NAME          VALUE
-----
lock_name_space      2
processes           150
sessions            247
timed_statistics    TRUE
timed_os_statistics 0
...

SQL> SHOW PARAMETER SHARED_POOL_SIZE
NAME          TYPE        VALUE
-----
shared_pool_size   big integer 0

SQL> show parameter para
NAME          TYPE        VALUE
-----
fast_start_parallel_rollback  string     LOW
parallel_adaptive_multi_user boolean   TRUE
parallel_automatic_tuning    boolean   FALSE
parallel_execution_message_size integer  16384
parallel_instance_group       string
...

```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### SQL\*Plus를 사용하여 파라미터 확인

이 슬라이드에서는 SQL\*Plus를 사용하여 파라미터를 확인하는 방법의 예제를 보여줍니다. 데이터 V\$PARAMETER 딕셔너리를 query하여 여러 파라미터의 값을 찾을 수 있습니다. V\$PARAMETER에는 현재 세션의 현재 파라미터 값이 표시됩니다. 또한 SHOW PARAMETER 명령에 원하는 문자열을 사용하여 해당 문자열을 포함하는 파라미터를 볼 수 있습니다. 다음 예제의 query는 파라미터의 이름 및 값을 요청합니다. 특정 파라미터 이름을 지정하려면 WHERE 절을 사용합니다.

```

SQL> SELECT name, value FROM V$PARAMETER WHERE name LIKE
  '%pool%';
NAME          VALUE
-----
shared_pool_size      0
large_pool_size       0
java_pool_size        0
streams_pool_size    0
shared_pool_reserved_size 8808038
buffer_pool_keep
...
9 rows selected.

```

## SQL\*Plus를 사용하여 파라미터 확인(계속)

뷰 설명:

Name	Null?	Type
NUM		NUMBER
NAME		VARCHAR2 (80)
TYPE		NUMBER
VALUE		VARCHAR2 (4000)
DISPLAY_VALUE		VARCHAR2 (4000)
ISDEFAULT		VARCHAR2 (9)
ISSES_MODIFIABLE		VARCHAR2 (5)
ISSYS_MODIFIABLE		VARCHAR2 (9)
ISINSTANCE_MODIFIABLE		VARCHAR2 (5)
ISMODIFIED		VARCHAR2 (10)
ISADJUSTED		VARCHAR2 (5)
ISDEPRECATED		VARCHAR2 (5)
ISBASIC		VARCHAR2 (5)
DESCRIPTION		VARCHAR2 (255)
UPDATE_COMMENT		VARCHAR2 (255)
HASH		NUMBER

두 번째 예제에서는 SQL\*Plus SHOW PARAMETER 명령을 사용하여 파라미터 설정을 보는 방법을 보여줍니다. 이 명령을 사용하여 텍스트 문자열을 포함하는 모든 파라미터를 찾을 수도 있습니다. 예를 들어, 다음 명령을 사용하면 db 문자열을 포함하는 모든 파라미터 이름을 찾을 수 있습니다.

NAME	TYPE	VALUE
...		
db_8k_cache_size	big integer	0
db_block_buffers	integer	0
db_block_checking	string	FALSE
db_block_checksum	string	TYPICAL
db_block_size	integer	8192
db_cache_advice	string	ON
db_cache_size	big integer	0
...		

파라미터에 대한 정보가 포함된 기타 뷰

- V\$SPPARAMETER: 서버 파라미터 파일 내용에 대한 정보가 표시됩니다. Instance를 시작하는 데 서버 파라미터 파일을 사용하지 않은 경우 이 뷰의 각 행에 있는 ISSPECIFIED 열에는 FALSE가 표시됩니다.
- V\$PARAMETER2: 현재 세션에 영향을 주는 초기화 파라미터에 대한 정보가 표시되며, 각 파라미터 값이 뷰의 행으로 표시됩니다. 새 세션은 V\$SYSTEM\_PARAMETER2 뷰에 표시되는 Instance 값에서 파라미터 값을 상속합니다.
- V\$SYSTEM\_PARAMETER: Instance에 현재 적용되고 있는 초기화 파라미터에 대한 정보를 표시합니다.

# 초기화 파라미터 값 변경

- **Static Parameter:**
  - 파라미터 파일에서만 변경 가능
  - Instance를 재시작해야 적용됨
  - 약 110개 정도의 파라미터
- **Dynamic Parameter:**
  - 데이터베이스가 온라인 상태인 동안 변경 가능
  - 다음 레벨에서 수정 가능
    - 세션 레벨
    - 시스템 레벨
  - 세션 기간 또는 SCOPE 설정에 따른 기간 동안 유효함
  - ALTER SESSION 및 ALTER SYSTEM 명령을 사용하여 변경
  - 약 234개 정도의 파라미터

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 초기화 파라미터 값 변경

다음과 같은 두 가지 유형의 초기화 파라미터가 있습니다.

**Static parameters:** Instance 또는 전체 데이터베이스에 영향을 주며, init.ora 또는 SPFILE의 내용을 변경해서만 수정할 수 있습니다. Static Parameter를 적용하려면 데이터베이스를 종료한 후에 재시작해야 합니다. 현재 Instance에 대해서는 Static Parameter를 변경할 수 없습니다.

**Dynamic Parameter:** 데이터베이스가 온라인 상태인 동안 변경할 수 있습니다. 이 파라미터에는 다음과 같은 두 가지 유형이 있습니다.

- **세션 레벨 파라미터**는 유저 세션에만 영향을 줍니다. 이에 대한 예로는 정렬, 날짜 파라미터 등의 국가별 언어 설정을 지정하는 데 사용할 수 있는 NLS(국가별 언어 지원) 파라미터가 있습니다. 이러한 파라미터는 지정된 세션에서 사용할 수 있으며 해당 세션이 끝나면 파라미터도 만료됩니다.
- **시스템 레벨 파라미터**는 전체 데이터베이스 및 모든 세션에 영향을 줍니다. 이에 대한 예로는 SGA\_TARGET 수정 및 아카이브 로그 대상 설정을 들 수 있습니다. 이러한 파라미터는 SCOPE 사양에 따라 효력이 유지됩니다. 이러한 파라미터를 영구 항목으로 만들려면 SCOPE=both 옵션을 지정하거나 수동으로 PFILE을 편집하여 해당 파라미터 설정을 SPFILE에 추가해야 합니다.

Dynamic Parameter는 ALTER SESSION 및 ALTER SYSTEM 명령을 사용하여 변경할 수 있습니다.

## 초기화 파라미터 값 변경(계속)

ALTER SYSTEM 문의 SET 절을 사용하여 초기화 파라미터 값을 설정하거나 변경하십시오. SCOPE 절(선택 사항)을 사용하여 다음과 같은 변경 범위를 지정할 수 있습니다.

- **SCOPE=SPFILE:** 변경 사항이 서버 파라미터 파일에만 적용되며 현재 Instance는 변경되지 않습니다. Dynamic Parameter와 Static Parameter 모두 변경 사항은 다음 시작 시에 영구적으로 적용됩니다. Static Parameter에 대해서는 이 SCOPE 사양만 허용됩니다.
- **SCOPE=MEMORY:** 변경 사항이 메모리에만 적용됩니다. 현재 Instance가 변경되고 변경 사항이 즉시 적용됩니다. Dynamic Parameter의 경우에는 서버 파라미터 파일이 생성되지 않으므로 변경 사항이 즉시 적용되기는 하지만 영구적으로 적용되지는 않습니다. Static Parameter의 경우에는 이 사양이 허용되지 않습니다.
- **SCOPE=BOTH:** 변경 사항이 서버 파라미터 파일과 메모리에 모두 적용됩니다. 현재 Instance가 변경되고 변경 사항이 즉시 적용됩니다. Dynamic Parameter의 경우에는 서버 파라미터 파일이 생성되므로 변경 사항이 영구적으로 적용됩니다. Static Parameter의 경우에는 이 사양이 허용되지 않습니다.

Instance를 시작하는 데 서버 파라미터 파일을 사용하지 않은 경우에는 SCOPE=SPFILE 또는 SCOPE=BOTH를 지정하면 오류가 발생합니다. 서버 파라미터 파일을 사용하여 Instance를 시작한 경우의 기본값은 SCOPE=BOTH이며, 텍스트 초기화 파라미터 파일을 사용하여 Instance를 시작한 경우의 기본값은 MEMORY입니다.

일부 dynamic parameter의 경우 DEFERRED 키워드도 지정할 수 있습니다. 이 키워드를 지정하는 경우 변경 사항은 이후 세션에 대해서만 적용됩니다. 이 키워드는 다음과 같은 파라미터에 대해서만 적합합니다.

- backup\_tape\_io\_slaves
- recyclebin
- audit\_file\_dest
- object\_cache\_optimal\_size
- object\_cache\_max\_size\_percent
- sort\_area\_size
- sort\_area\_retained\_size
- olap\_page\_pool\_size

SCOPE를 SPFILE 또는 BOTH로 지정하면 선택 사항인 COMMENT 절을 통해 텍스트 문자열을 파라미터 생성과 연관시킬 수 있습니다. 그러면 서버 파라미터 파일에 주석이 기록됩니다.

## 파라미터 값 변경: 예제

```
SQL> ALTER SESSION  
      SET NLS_DATE_FORMAT = 'mon dd yyyy';  
  
Session altered.  
  
SQL> SELECT SYSDATE FROM dual;  
  
SYSDATE  
-----  
jun 18 2009
```

```
SQL> ALTER SYSTEM SET  
SEC_MAX_FAILED_LOGIN_ATTEMPTS=2 COMMENT='Reduce  
from 10 for tighter security.' SCOPE=SPFILE;  
  
System altered.
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 파라미터 값 변경: 예제

슬라이드의 첫번째 명령문은 세션 레벨 파라미터를 변경하는 예제입니다. 유저는 세션 날짜 형식을 mon dd yyyy로 설정합니다. 그러면 해당 날짜의 모든 query에 날짜가 해당 형식으로 표시됩니다. 세션 레벨 파라미터는 PL/SQL을 사용하여 응용 프로그램에서 설정할 수도 있습니다.

두번째 명령문에서는 연결이 끊기기 전에 실패한 로그인 시도의 최대 수를 변경합니다. 이 명령문은 주석을 포함하며, 서버 파라미터 파일에서만 변경을 수행하도록 명시적으로 지정합니다. 지정된 Failure 시도 수를 초과하면 서버 프로세스가 연결을 자동으로 끊습니다. 이는 Dynamic Parameter가 아니며 변경 내용을 적용하려면 오라클 데이터베이스 Instance를 재시작해야 합니다.

## 퀴즈

Enterprise Manager Database Control을 사용하면 여러 데이터베이스를 동시에 관리할 수 있습니다.

1. 참
2. 거짓

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 퀴즈

데이터베이스 파라미터의 대부분은 동적이며, 데이터베이스 Instance를 종료하지 않고도 변경할 수 있습니다.

1. 참
2. 거짓

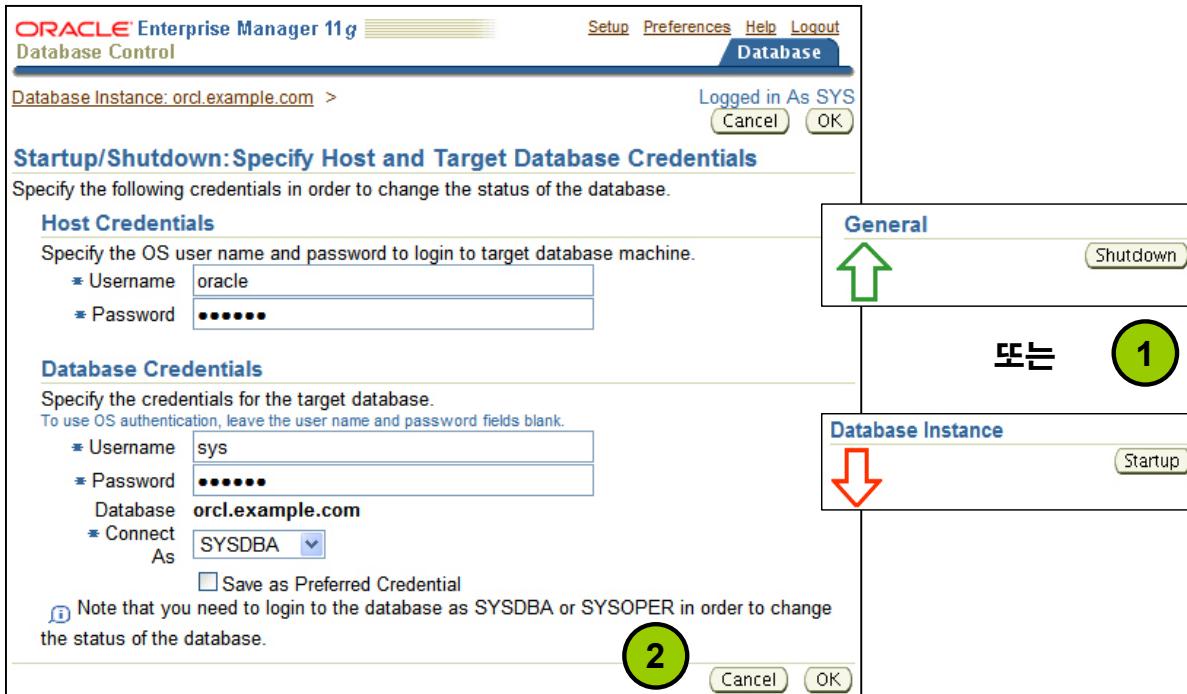
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1

# 데이터베이스 시작 및 종료: 인증서

구성 요소  
SQL\*Plus  
초기화 파라미터  
> DB 시작 DB 종료  
Alert Log  
성능 뷰



ORACLE

Copyright © 2010, Oracle. All rights reserved.

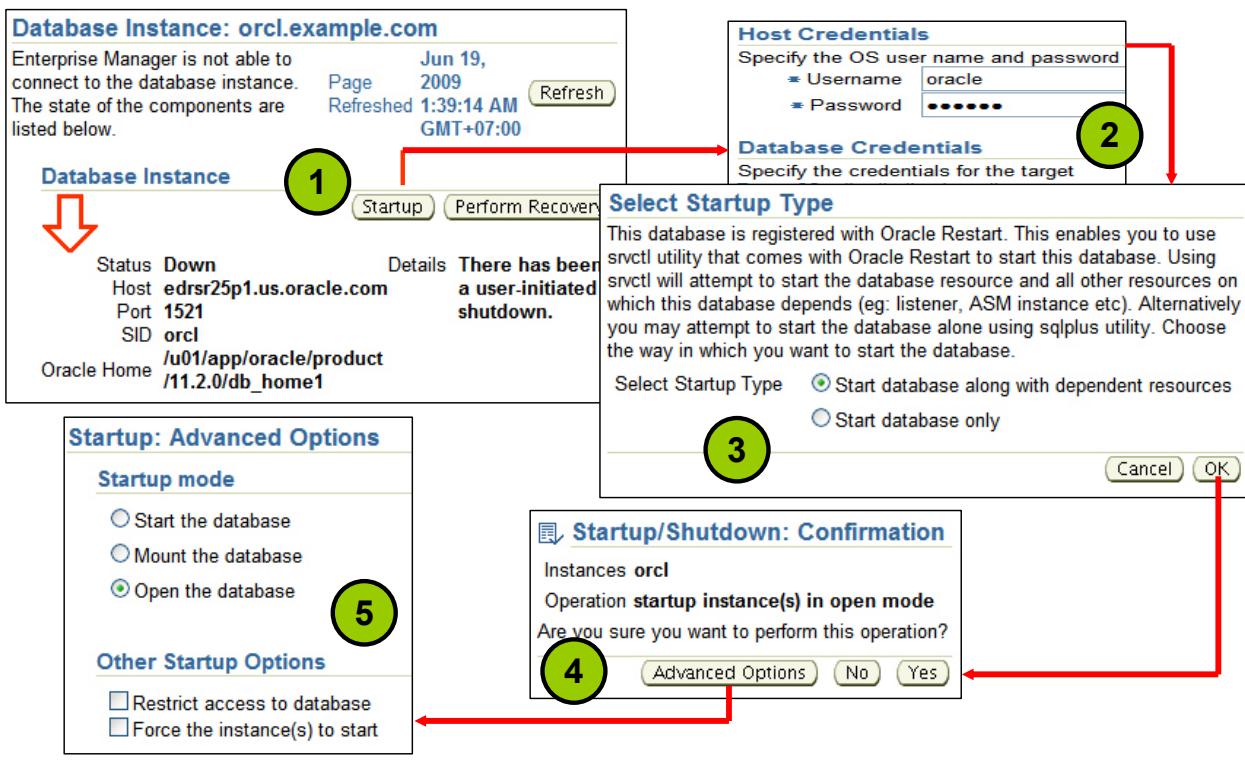
## 데이터베이스 시작 및 종료: 인증서

Startup 또는 Shutdown을 누르면 호스트(데이터베이스가 있는 컴퓨터)와 데이터베이스 자체에 로그인하는 데 사용할 인증서를 지정하라는 메시지가 나타납니다. SYSDBA 권한이 있는 데이터베이스 계정을 입력해야 합니다. 인증서를 입력합니다.

인증서 정보를 입력하면 시작 또는 종료 방법을 선택하라는 메시지가 표시됩니다. 필요한 경우 Advanced Options를 눌러 시작 옵션 또는 종료 모드를 변경할 수 있습니다. 또한 Show SQL을 눌러 시작 또는 종료에 사용되는 SQL 문을 확인할 수 있습니다.

**주:** EM을 사용한 기본 종료 옵션은 IMMEDIATE입니다. 그리고 SQL\*Plus에서 SHUTDOWN 명령을 실행할 때의 기본 값은 NORMAL입니다.

# 오라클 데이터베이스 Instance 시작



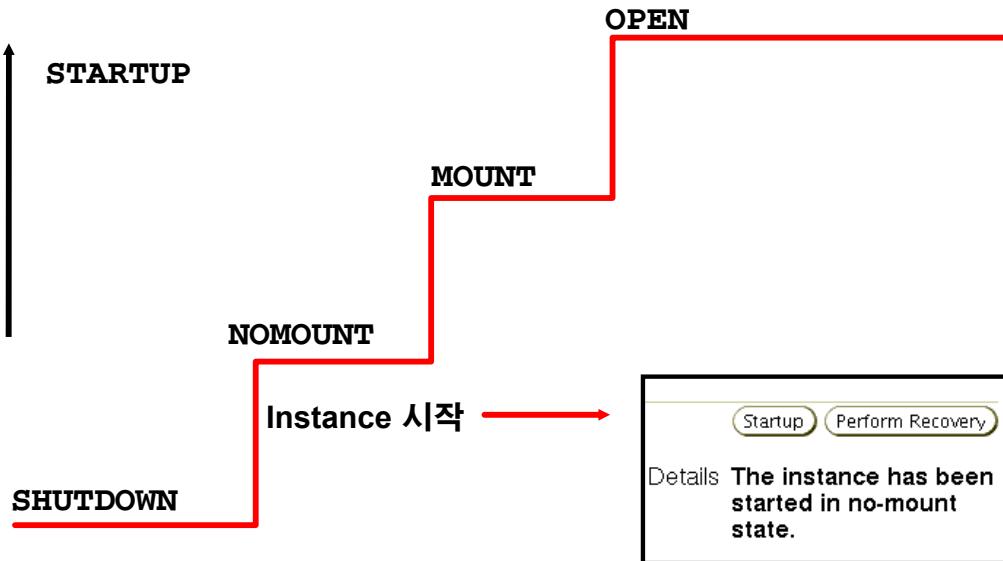
ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스 Instance 시작

데이터베이스가 현재 시작되어 있지 않은 경우 Enterprise Manager Database Control 페이지에서 Startup을 누릅니다. 호스트 인증서(Host Credentials)를 입력하고 선택적으로 시작 모드를 선택합니다. 오라클 데이터베이스가 Oracle Restart에 등록된 경우, 데이터베이스 Instance를 시작할 때 Server Control(SRVCTL) 유ти리티 또는 SQL\*Plus를 사용할지를 묻는 별도의 대화상자가 표시됩니다. Oracle Restart를 사용할 때는 필요한 종속 리소스를 시작할 수 있는 기능이 있는 SRVCTL 유ти리티가 권장 유ти리티입니다.

# 오라클 데이터베이스 Instance 시작: NOMOUNT



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스 Instance 시작: NOMOUNT

데이터베이스 instance를 시작할 때는 시작 상태를 선택합니다. 다음 시나리오에서는 instance를 시작하는 여러 단계에 대해 설명합니다.

일반적으로 데이터베이스 생성, 콘트롤 파일 재생성 또는 특정 백업 및 recovery 시나리오 실행 중에는 instance가 NOMOUNT 모드에서만 시작됩니다.

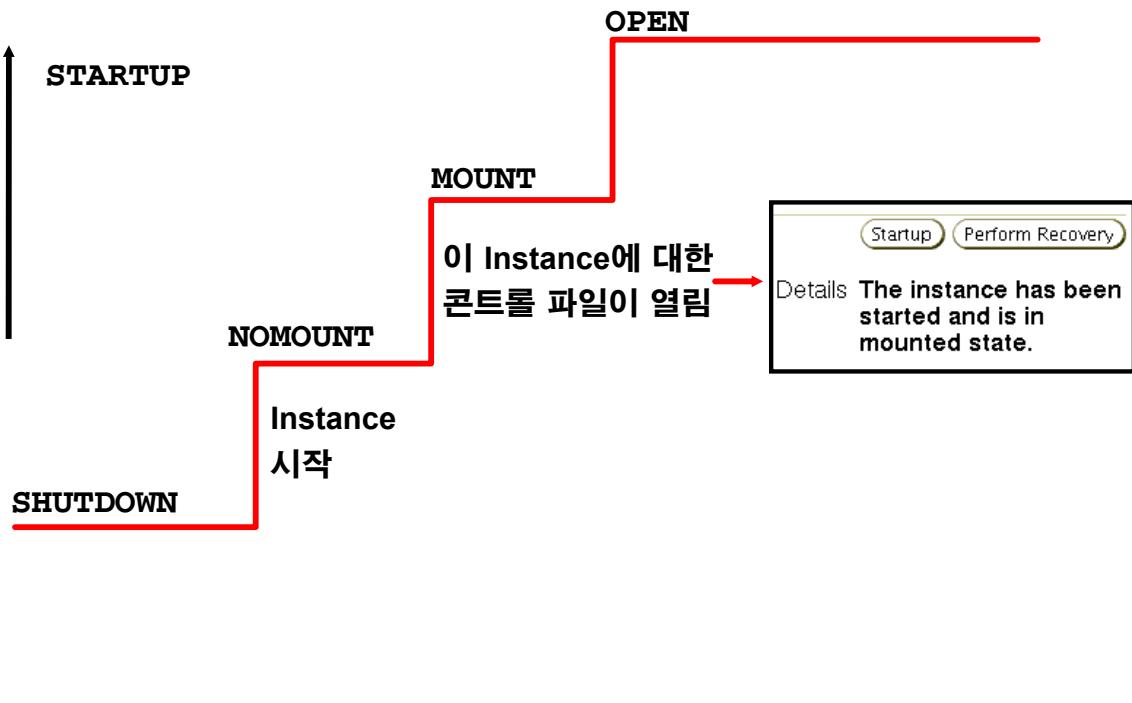
Instance 시작에는 다음 작업이 포함됩니다.

- 다음 순서로 \$ORACLE\_HOME/dbs에서 특정 이름의 파일 검색
  - spfile<SID>.ora를 검색합니다.
  - spfile<SID>.ora가 없으면 spfile.ora를 검색합니다.
  - spfile.ora가 없으면 init<SID>.ora를 검색합니다.

이 파일은 instance의 초기화 파라미터가 들어 있는 파일입니다. STARTUP에서 PFILE 파라미터를 지정하면 기본값 대신 이 파라미터가 적용됩니다.
- SGA 할당
- 백그라운드 프로세스 시작
- alert\_<SID>.log 파일 및 trace file 열기

주: SID는 instance를 식별하는 시스템 ID입니다(예: ORCL).

## 오라클 데이터베이스 Instance 시작: MOUNT



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 오라클 데이터베이스 Instance 시작: MOUNT

데이터베이스 마운트에는 다음 작업이 포함됩니다.

- 이전에 시작된 Instance와 데이터베이스 연관
- 파라미터 파일에 지정된 콘트롤 파일을 찾아서 열기
- 데이터 파일과 온라인 리두 로그 파일의 이름 및 상태를 알기 위해 콘트롤 파일 읽기 (그러나 이때 데이터 파일과 온라인 리두 로그 파일의 존재 여부를 확인하는 검사는 수행하지 않습니다.)

특정 유지 관리 작업을 수행하려면 instance를 시작하고 데이터베이스를 마운트하는 단계까지만 수행하고 데이터베이스를 열지는 마십시오.

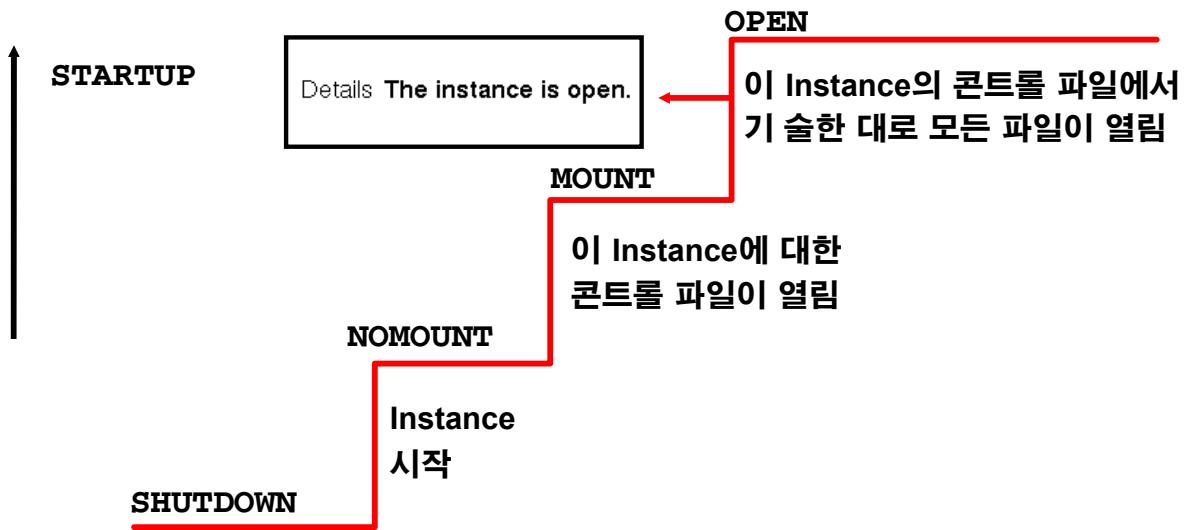
예를 들어, 다음 작업을 수행하는 동안에는 데이터베이스가 마운트되어 있어야 하지만 열려 있으면 안됩니다.

- 데이터 파일 이름 바꾸기(오프라인 테이블스페이스에 대한 데이터 파일의 이름은 데이터베이스가 열린 상태에서도 바꿀 수 있음)
- 온라인 리두 로그 파일 아카이브 옵션 활성화 및 비활성화
- 전체 데이터베이스 Recovery 수행

주: OPEN 요청이 수행된 이후에도 데이터베이스가 MOUNT 모드로 남아 있을 수 있습니다.

그 이유는 데이터베이스를 일정 방식으로 recovery할 필요가 있기 때문일 수 있습니다. MOUNT 상태에서 Recovery를 수행하면 리두 로그가 읽기용으로 열리고, Recovery가 필요한 블록을 읽고 Recovery 중에 필요한 경우 블록을 기록하기 위해 데이터 파일도 열립니다.

## 오라클 데이터베이스 Instance 시작: OPEN



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 오라클 데이터베이스 Instance 시작: OPEN

정상적인 데이터베이스 운영 상태는 instance가 시작되고 데이터베이스가 마운트되고 열린 상태입니다. 정상적인 데이터베이스 운영 상태에서는 유효한 모든 유저가 데이터베이스에 연결하여 일반적인 데이터 액세스 작업을 수행할 수 있습니다.

데이터베이스 열기에는 다음 작업이 포함됩니다.

- 데이터 파일 열기
- 온라인 리두 로그 파일 열기

데이터베이스를 열 때 데이터 파일이나 온라인 리두 로그 파일이 없으면 Oracle 서버는 오류를 반환합니다.

이 최종 단계에서 Oracle 서버는 모든 데이터 파일 및 온라인 리두 로그 파일을 열 수 있는지 확인하고 데이터베이스의 일관성을 검사합니다. 필요한 경우 시스템 모니터(SMON) 백그라운드 프로세스가 instance recovery를 시작합니다.

관리 권한을 가진 유저만 instance를 사용할 수 있는 제한 모드로 데이터베이스 instance를 시작할 수 있습니다. 제한 모드로 Instance를 시작하려면 Advanced Startup Options 페이지에서 "Restrict access to database" 옵션을 선택합니다.

## 시작 옵션: 예제

- sqlplus 유ти리티 사용:

```
SQL> startup
```

1

```
SQL> startup nomount
```

2

```
SQL> alter database mount;
```

3

```
SQL> alter database open;
```

4

- Oracle Restart에서 srvctl 유ти리티 사용

```
$ srvctl start database -d orcl -o mount
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 시작 옵션: 예제

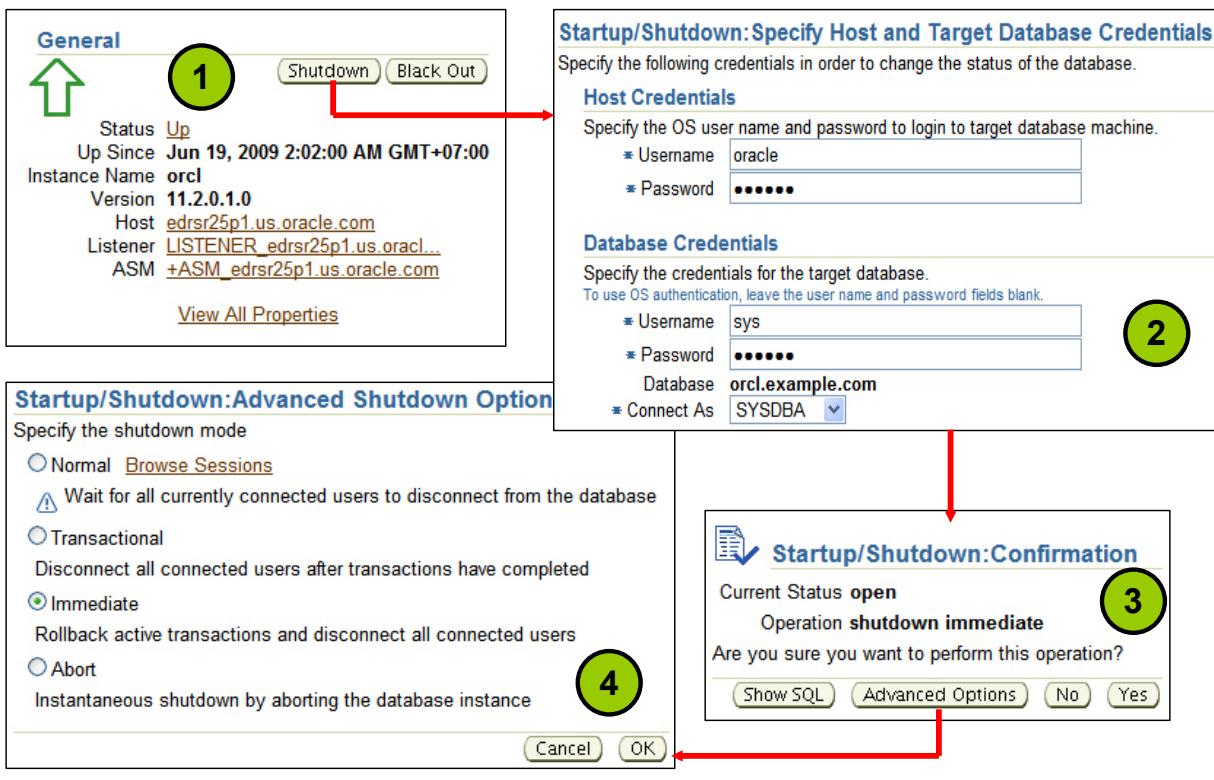
이 슬라이드에서는 데이터베이스를 시작하기 위한 SQL\*Plus 구문을 보여줍니다.

- 이 명령은 Instance를 시작하고, Instance에 데이터베이스 파일을 연관시키고, 데이터베이스를 마운트하고 엽니다.
- 이 명령은 Instance를 시작합니다. 데이터베이스는 마운트되지 않습니다.
- 이 명령은 NOMOUNT 상태에서 데이터베이스를 마운트합니다.
- 이 명령은 MOUNT 상태에서 데이터베이스를 엽니다.

데이터베이스가 Oracle Restart에 활성화된 경우 srvctl 유ти리티를 사용하여 데이터베이스 Instance를 시작할 수 있습니다. srvctl 유ти리티는 ASM Instance, ASM 디스크 그룹 및 리스너와 같은 모든 필수 종속 리소스를 시작할 수도 있다는 장점이 있습니다.

주: srvctl 유ти리티는 그리드 Infrastructure 소프트웨어를 위한 \$ORACLE\_HOME/bin 디렉토리와 오라클 데이터베이스 소프트웨어를 위한 \$ORACLE\_HOME/bin 디렉토리에 있습니다. 오라클 데이터베이스를 시작할 때는 오라클 데이터베이스 소프트웨어에서 srvctl 유ти리티를 사용해야 합니다. ASM Instance 또는 리스너를 시작할 때는 그리드 Infrastructure 소프트웨어에서 srvctl 유ти리티를 사용해야 합니다.

# 오라클 데이터베이스 Instance 종료



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 오라클 데이터베이스 Instance 종료

Instance가 이미 시작된 경우 Enterprise Manager Database Control 페이지에서 Shutdown 버튼을 눌러 Instance를 종료합니다. 호스트 및 데이터베이스 인증서를 확인 또는 입력하라는 메시지가 표시됩니다. Startup/Shutdown 확인 대화상자에서 OK를 누릅니다. Advanced Options 버튼을 누르면 종료 모드로 NORMAL, TRANSACTIONAL, IMMEDIATE 또는 ABORT를 선택할 수 있습니다.

## 종료 모드

종료 모드	A	I	T	N
새로운 연결 허용	아니오	아니오	아니오	아니오
현재 세션 종료 시까지 대기	아니오	아니오	아니오	예
현재 트랜잭션 종료 시까지 대기	아니오	아니오	예	예
체크포인트 적용 및 파일 닫기	아니오	예	예	예

### 종료 모드:

- **A = ABORT**
- **I = IMMEDIATE**
- **T = TRANSACTIONAL**
- **N = NORMAL**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 종료 모드

종료 모드는 다음과 같은 순서로 갈수록 더욱 적극적으로 현재 작업을 조정하게 됩니다.

- **ABORT:** 종료하기 전에 최소한의 작업을 수행합니다. 이 모드를 사용하면 시작하기 전에 Recovery가 수행되므로 꼭 필요한 경우에만 사용해야 합니다. 이 모드는 일반적으로 다른 형태의 종료 모드가 작동하지 않거나 Instance를 시작할 때 문제가 있는 경우 또는 수초 이내에 정전이 된다는 통보가 있는 경우와 같은 급박한 상황으로 인해 즉시 종료해야 하는 경우에 사용됩니다.
- **IMMEDIATE:** 가장 일반적으로 사용되는 옵션입니다. 커밋되지 않은 트랜잭션은 롤백됩니다.
- **TRANSACTIONAL:** 기존 트랜잭션을 종료할 수 있지만 새 트랜잭션을 시작하지 않습니다.
- **NORMAL:** 세션의 연결이 끊길 때까지 대기합니다.

종료하는 데 걸리는 시간은 ABORT가 가장 빠르고 NORMAL이 가장 느립니다. NORMAL 및 TRANSACTIONAL은 세션 및 트랜잭션 수에 따라 시간이 오래 걸릴 수 있습니다.

# 종료 옵션

## 아래쪽 방향:

- IMMEDIATE의 경우 커밋되지 않은 변경 내용이 롤백됨
- 데이터베이스 버퍼 캐시가 데이터 파일에 기록됨
- 리소스 해제

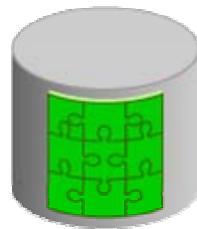
## 아래 기간 중:

SHUTDOWN  
NORMAL  
또는  
SHUTDOWN  
TRANSACTIONAL  
또는  
SHUTDOWN  
IMMEDIATE

## 위쪽 방향:

- Instance Recovery 없음

일관성 있는 데이터베이스



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 종료 옵션

### SHUTDOWN NORMAL

모드가 지정되어 있지 않을 경우 NORMAL 모드가 기본값입니다. Normal 데이터베이스 종료 모드에서는 다음과 같은 조건이 만족됩니다.

- 새 연결을 생성할 수 없습니다.
- Oracle 서버는 모든 유저가 연결을 끊을 때까지 종료하지 않고 대기합니다.
- 데이터베이스 및 리두 버퍼가 디스크에 기록됩니다.
- 백그라운드 프로세스가 종료되고 SGA가 메모리에서 제거됩니다.
- Oracle 서버는 Instance를 종료하기 전에 데이터베이스를 닫고 마운트를 해제합니다.
- 이후 시작 시 Instance를 Recovery할 필요가 없습니다.

### SHUTDOWN TRANSACTIONAL

TRANSACTIONAL 종료 모드를 사용하면 클라이언트의 현재 작업 결과를 포함하여 데이터가 손실되지 않습니다. Transactional 데이터베이스 종료 모드에서는 다음과 같은 조건이 만족됩니다.

- 해당 Instance에서 클라이언트가 새 트랜잭션을 시작할 수 없습니다.
- 진행 중인 트랜잭션이 종료되면 클라이언트의 연결이 해제됩니다.
- 모든 트랜잭션이 완료되는 즉시 데이터베이스가 종료됩니다.
- 이후 시작 시 Instance를 Recovery할 필요가 없습니다.

## 종료 옵션(계속)

### SHUTDOWN IMMEDIATE

IMMEDIATE 종료 모드에서는 다음과 같은 조건이 만족됩니다.

- 현재 오라클 데이터베이스가 처리하고 있는 SQL 문이 완료되지 않습니다.
- Oracle 서버는 현재 데이터베이스에 연결하고 있는 유저의 연결이 해제될 때까지 대기하지 않습니다.
- Oracle 서버는 활성 트랜잭션을 롤백하고 모든 유저의 연결을 해제합니다.
- Oracle 서버는 Instance를 종료하기 전에 데이터베이스를 닫고 마운트를 해제합니다.
- 이후 시작 시 instance를 recovery할 필요가 없습니다.

주: IMMEDIATE는 Enterprise Manager를 사용할 때의 기본 종료 모드입니다.

## 종료 옵션

### 아래쪽 방향:

- 수정된 버퍼가 데이터 파일에 기록되지 않음
- 커밋되지 않은 변경 내용이 롤백되지 않음



### 아래 기간 중:

SHUTDOWN ABORT  
또는  
Instance Failure  
또는  
STARTUP FORCE

### 위쪽 방향:

- 온라인 리두 로그 파일을 사용하여 변경 내용이 재적용됨
- 언두 세그먼트가 커밋되지 않은 변경 사항을 롤백하는 데 사용
- 리소스 해제

일관성 없는 데이터베이스

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 종료 옵션(계속)

#### SHUTDOWN ABORT

NORMAL, TRANSACTIONAL 및 IMMEDIATE 종료 모드가 작동하지 않는 경우 현재 데이터베이스 Instance를 중지할 수 있습니다. Instance 중지 모드에서는 다음과 같은 조건이 만족됩니다.

- 현재 Oracle 서버가 처리하고 있는 SQL 문이 즉시 종료됩니다.
- Oracle 서버는 현재 데이터베이스에 연결하고 있는 유저의 연결이 해제될 때까지 대기하지 않습니다.
- 데이터베이스 및 리두 버퍼가 디스크에 기록되지 않습니다.
- 커밋되지 않은 트랜잭션은 롤백되지 않습니다.
- 파일을 닫지 않은 상태로 instance가 종료됩니다.
- 데이터베이스가 닫히거나 마운트 해제되지 않습니다.
- 다음 번 시작 시 Instance Recovery가 필요하며 이는 자동으로 수행됩니다.

주: 일관성 없는 상태의 데이터베이스는 백업하지 않는 것이 좋습니다.

## 종료 옵션: 예제

- SQL\*Plus 사용:

```
SQL> shutdown
```

1

```
SQL> shutdown transactional
```

2

```
SQL> shutdown immediate
```

3

```
SQL> shutdown abort
```

4

- Oracle Restart에서 SRVCTL 유ти리티 사용

```
$ srvctl stop database -d orcl -o abort
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 종료 옵션: 예제

이 슬라이드에서는 SQL\*Plus 및 SRVCTL 유ти리티를 모두 사용하여 데이터베이스를 종료하는 방법의 예제를 보여 줍니다.

- 이 명령은 정상 종료를 시작합니다. 데이터베이스는 모든 유저가 로그아웃할 때까지 종료되지 않습니다.
- 이 명령은 트랜잭션 종료를 시작합니다. 데이터베이스는 모든 기존 트랜잭션이 완료될 때까지 종료되지 않습니다.
- 이 명령은 즉시 종료를 시작합니다. 커밋되지 않은 트랜잭션은 롤백됩니다.
- 이 명령은 종료 중지를 시작합니다.

데이터베이스가 Oracle Restart에 활성화된 경우 SRVCTL 유ти리티를 사용하여 데이터베이스 Instance를 종료할 수 있습니다.

**주:** SRVCTL 유ти리티는 그리드 Infrastructure 소프트웨어를 위한 \$ORACLE\_HOME/bin 디렉토리와 오라클 데이터베이스 소프트웨어를 위한 \$ORACLE\_HOME/bin 디렉토리에 있습니다. 오라클 데이터베이스를 시작할 때는 오라클 데이터베이스 소프트웨어에서 SRVCTL 유ти리티를 사용해야 합니다. ASM Instance 또는 리스너를 시작할 때는 그리드 Infrastructure 소프트웨어에서 SRVCTL 유ти리티를 사용해야 합니다.

# Alert Log 확인

Database Home 페이지 > Related Links 영역 >  
Alert Log Content

구성 요소  
SQL\*Plus  
초기화 파라미터  
DB 시작  
DB 종료  
> Alert Log  
성능 뷰

View Entries Last 50 Go Search						
Timestamp	Type ▲	Level	Incident ID	Group	Message ID	Message Text
Jun 19, 2009 10:00:16 PM GMT+07:00	NOTIFICATION	16		sqltune	kesaiTuneSqlDrv:5067:3456118459	End automatic SQL Tuning Advisor run for special tuning task "SYS_AUTO_SQL_TUNING_TASK"
Jun 19, 2009 10:00:03 PM GMT+07:00	NOTIFICATION	16		sqltune	kesaiTuneSqlDrv:4555:2579917519	Begin automatic SQL Tuning Advisor run for special tuning task "SYS_AUTO_SQL_TUNING_TASK"
Jun 19, 2009 10:00:00 PM GMT+07:00	NOTIFICATION	16		process start	ksbrdp:3833:3697353022	VKRM started with pid=24, OS id=7929
Jun 19, 2009 10:00:00 PM GMT+07:00	NOTIFICATION	16		process start	ksbs1p_real:2253:2371767696	Starting background process VKRM
Jun 19, 2009 2:07:22 AM GMT+07:00	NOTIFICATION	16		process start	ksbrdp:3833:3697353022	SMCO started with pid=23, OS id=30582
Jun 19, 2009 2:07:22 AM GMT+07:00	NOTIFICATION	16		process start	ksbs1p_real:2253:2371767696	Starting background process SMCO
Jun 19, 2009 2:02:26 AM GMT+07:00	NOTIFICATION	16		process start	ksbrdp:3833:3697353022	CJQ0 started with pid=33, OS id=29846

Copyright © 2010, Oracle. All rights reserved.

## Alert Log 확인

각 데이터베이스에는 alert\_<sid>.log 파일이 있습니다. 이 파일은 데이터베이스가 포함된 서버에 있으며, \$ORACLE\_BASE가 설정된 경우 기본적으로 \$ORACLE\_BASE/diag/rdbms/<db\_name>/<SID>/trace에 저장됩니다.

데이터베이스의 Alert File에는 다음과 같은 메시지가 시간 순으로 기록되어 있습니다.

- 시작 시 사용된 기본값이 아닌 초기화 파라미터
- 발생한 모든 내부 오류(ORA-600), 블록 손상 오류(ORA-1578) 및 deadlock 오류(ORA-60)
- CREATE, ALTER, DROP DATABASE 및 TABLESPACE SQL 문 및 STARTUP, SHUTDOWN, ARCHIVE LOG 및 RECOVER와 같은 Enterprise Manager 또는 SQL\*Plus 문과 같은 관리 작업
- shared server 및 디스패처 프로세스의 기능과 관련된 일부 메시지 및 오류
- Materialized View의 자동 Refresh 중에 발생한 오류

오라클 데이터베이스는 Alert Log를 사용하여 운영자의 콘솔에 정보를 표시하는 대체 수단으로 이러한 이벤트 레코드를 보관합니다. 대부분의 시스템에서는 콘솔에도 이 정보가 표시됩니다. 관리 작업을 성공적으로 수행하면 Alert Log에 시간 기록과 함께 "completed"라는 메시지가 기록됩니다.

## Alert Log 확인(계속)

Enterprise Manager는 Alert Log 파일을 모니터하고 critical 오류를 유저에게 알려줍니다. 로그에서 non-critical 오류 및 정보 메시지도 확인할 수 있습니다. 파일 크기가 너무 커져서 관리할 수 없는 경우도 있으므로 Alert 파일을 주기적으로 백업하고 현재 Alert 파일을 삭제하는 것이 좋습니다. 데이터베이스가 Alert 파일에 다시 쓰려고 시도하면 새 파일이 생성됩니다.

**주:** \$ORACLE\_BASE/diag/rdbms/<db\_name>/<SID>/alert 디렉토리에는 XML 버전의 Alert Log가 있습니다.

### SQL\*Plus로 Alert Log의 위치를 확인하려면 다음을 수행하십시오.

- SQL\*Plus 또는 SQL Developer 등의 다른 query 도구를 사용하여 데이터베이스에 연결합니다.
- V\$DIAG\_INFO 뷰를 query합니다.

### XML 태그가 없는 텍스트 전용 Alert Log를 보려면 다음을 수행하십시오.

- V\$DIAG\_INFO query 결과에서 Diag Trace 항목에 해당하는 경로를 기록해둡니다. 디렉토리를 해당 경로로 변경합니다.
- 텍스트 편집기를 사용하여 alert\_SID.log 파일을 엽니다.

### XML 형식 Alert Log를 보려면 다음을 수행하십시오.

- V\$DIAG\_INFO query 결과에서 Diag Alert 항목에 해당하는 경로를 기록해둡니다. 디렉토리를 해당 경로로 변경합니다.
- 텍스트 편집기를 사용하여 log.xml 파일을 엽니다.

## Trace file 사용

- 각 서버 프로세스와 백그라운드 프로세스는 연관된 Trace file에 정보를 기록할 수 있습니다.
- 오류 정보는 해당하는 Trace file에 기록됩니다.
- ADR(Automatic Diagnostic Repository)
  - 시스템 중앙 추적 및 로깅 Repository입니다.
  - 다음과 같은 데이터베이스 진단 데이터를 저장합니다.
    - Trace
    - Alert log
    - 상태 모니터 보고서

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

### Trace file 사용

각 서버 프로세스와 백그라운드 프로세스는 연관된 Trace file에 정보를 기록할 수 있습니다. 프로세스는 시스템 오류를 감지하면 오류에 대한 정보를 해당 Trace file로 덤프합니다. 시스템 오류가 발생하고 정보가 Trace file에 기록되면 관리자는 Oracle 지원 센터 서비스에 문의해야 합니다.

백그라운드 프로세스와 연관된 Trace file의 모든 파일 이름에는 Trace file을 생성한 프로세스의 이름이 포함되어 있습니다. 그러나 작업 큐 프로세스(Jnnn)에 의해 생성되는 Trace file의 경우는 예외입니다.

Trace file의 추가 정보를 통해 응용 프로그램 또는 Instance 튜닝을 위한 지침을 확인할 수 있습니다. 백그라운드 프로세스는 적합한 경우 항상 이 정보를 Trace file에 기록합니다.

Oracle Database 11g부터는 문제 방지, 감지, 진단 및 해결을 위한 고급 오류 진단 기능 Infrastructure가 포함되었습니다. 특히 이 Infrastructure의 대상이 되는 문제로는 데이터베이스 코드 버그, 메타데이터 손상, 고객 데이터 손상 등으로 인해 발생하는 critical 오류가 포함됩니다.

## Trace file 사용(계속)

critical 오류가 발생하면 incident 번호가 오류에 할당되고 Trace file 등 오류의 진단 데이터가 즉시 캡처되며 이 번호로 태그가 지정됩니다. 그런 다음 데이터는 데이터베이스 외부에 있는 파일 기반 Repository인 ADR(Automatic Diagnostic Repository)에 저장되므로 나중에 incident 번호별로 검색하여 분석할 수 있습니다.

ADR은 Trace, Alert Log, 상태 모니터 보고서 등 데이터베이스 진단 데이터를 저장하는 데 사용하는 시스템 중앙 추적 및 로깅 Repository입니다.

ADR 루트 디렉토리는 *ADR* 기준이라고도 하며, 해당 위치는 DIAGNOSTIC\_DEST 초기화 파라미터에 의해 설정됩니다. 이 파라미터를 생략하거나 null로 유지하면 데이터베이스가 다음과 같이 시작 시에 DIAGNOSTIC\_DEST를 설정합니다.

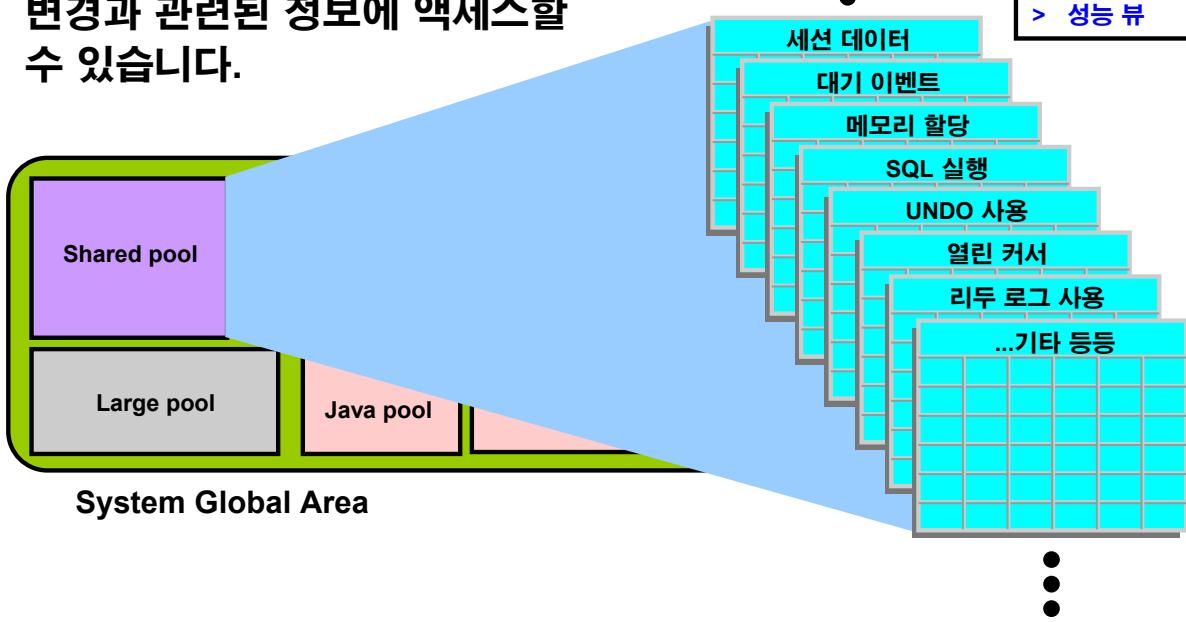
- ORACLE\_BASE 환경 변수를 설정하는 경우 DIAGNOSTIC\_DEST는 ORACLE\_BASE가 지정하는 디렉토리로 설정됩니다.
- ORACLE\_BASE 환경 변수를 설정하지 않은 경우 DIAGNOSTIC\_DEST는 ORACLE\_HOME/log로 설정됩니다.

ADR 흄의 위치는 ADR 기준 디렉토리에서 시작되는 다음 경로에 의해 지정됩니다.

```
./diag/product_type/db_id/instance_id
```

# Dynamic Performance 뷰

Instance 메모리 구조의 상태  
변경과 관련된 정보에 액세스할  
수 있습니다.



구성 요소  
SQL\*Plus  
초기화 파라미터  
DB 시작  
DB 종료  
Alert Log  
> 성능 뷰

Copyright © 2010, Oracle. All rights reserved.

ORACLE

## Dynamic Performance 뷰

오라클 데이터베이스는 데이터베이스 instance의 작업 및 성능에 대한 동적 데이터 집합도 유지 관리합니다. 이러한 Dynamic Performance 뷰는 데이터베이스 서버 내의 메모리 구조로부터 작성된 가상 테이블을 기반으로 합니다. 즉, 데이터베이스에 있는 일반 테이블이 아니므로, 데이터베이스가 마운트되거나 열리기 전에도 일부 뷰를 사용할 수 있습니다.

Dynamic Performance 뷰에는 다음 정보가 포함됩니다.

- 세션
- 파일 상태
- 작업 진행 상황
- Lock
- 백업 상태
- 메모리 사용 및 할당
- 시스템 및 세션 파라미터
- SQL 실행
- 통계 및 Metric

주: DICT 및 DICT\_COLUMNS 뷰에는 이러한 Dynamic Performance 뷰의 이름도 포함됩니다.  
Dynamic Performance 뷰는 접두어 'v\$'로 시작하며 590개 이상 있습니다.

## Dynamic Performance 뷰: 사용 예제

1

```
SQL> SELECT sql_text, executions FROM v$sql  
WHERE cpu_time > 200000;
```

2

```
SQL> SELECT * FROM v$session WHERE machine =  
'EDRSR9P1' and logon_time > SYSDATE - 1;
```

3

```
SQL> SELECT sid, ctime FROM v$lock  
WHERE block > 0;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Dynamic Performance 뷰: 사용 예제

이 뷰는 주로 Enterprise Manager에서 사용하지만 일반 유저도 필요한 경우 이 뷰를 query 할 수 있습니다. 슬라이드에 있는 세 가지 예는 다음 질문에 대한 답을 제공합니다.

1. CPU 시간이 200,000마이크로초 이상 소비되는 SQL 문은 무엇이고 연관된 실행 횟수는 몇 번입니까?
2. 전날 EDRSR9P1 컴퓨터에서 로그인한 현재 세션은 어느 것입니까?
3. 현재 다른 유저를 차단 중인 Lock을 보유하고 있는 세션의 세션 ID는 무엇이며 Lock 상태가 얼마 동안 유지되고 있습니까?

## Dynamic Performance 뷰: 고려 사항

- 이러한 뷰는 sys 유저가 소유합니다.
- 사용 시점에 따라 서로 다른 뷰를 사용할 수 있습니다.
  - Instance가 시작된 경우
  - 데이터베이스가 마운트된 경우
  - 데이터베이스가 열린 경우
- V\$FIXED\_TABLE을 query하면 모든 뷰 이름을 볼 수 있습니다.
- 이 뷰를 "v-dollar 뷰"라고도 부릅니다.
- 데이터가 동적이기 때문에 이 뷰에서는 읽기 일관성이 보장되지 않습니다.

ORACLE®

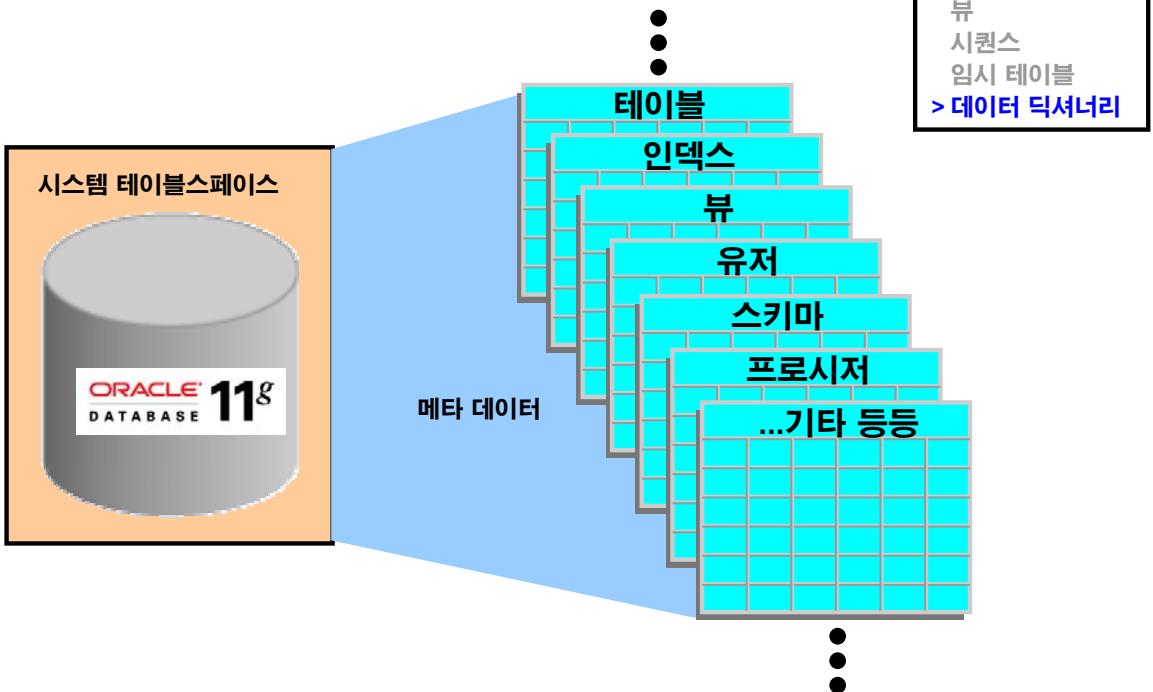
Copyright © 2010, Oracle. All rights reserved.

### Dynamic Performance 뷰: 고려 사항

일부 동적 뷰는 Instance 또는 데이터베이스의 일부 상태에 적용할 수 없는 데이터가 포함됩니다. 예를 들어, Instance가 방금 시작되었지만 데이터베이스가 마운트되지 않은 경우 V\$BGPROCESS를 query하여 실행 중인 백그라운드 프로세스 리스트를 볼 수 있습니다. 하지만 V\$DATAFILE을 query하여 데이터베이스 데이터 파일의 상태를 볼 수는 없습니다. 왜냐하면 콘트롤 파일을 읽어서 데이터베이스와 연결된 데이터 파일을 찾는 작업은 데이터베이스 마운트 과정에서 수행되기 때문입니다.

일부 V\$ 뷰에는 해당하는 DBA\_ 뷰의 정보와 유사한 정보가 포함되어 있습니다. 예를 들어, V\$DATAFILE은 DBA\_DATA\_FILES와 비슷합니다. 또한 V\$ 뷰 이름은 일반적으로 단수이며 DBA\_ 뷰 이름은 복수입니다.

# 데이터 딕셔너리: 개요



```
SELECT * FROM dictionary;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 데이터 딕셔너리: 개요

오라클 데이터 딕셔너리는 데이터베이스에 대한 메타 데이터이며 데이터베이스에 있는 모든 객체의 이름과 속성을 포함합니다. 객체를 생성하거나 수정하면 해당 변경 사항을 반영하는 데이터 딕셔너리가 갱신됩니다. 이 정보는 오라클 데이터베이스에 의해 유지 관리되는 기본 테이블(base table)에 저장되지만 테이블을 직접 읽는 대신 미리 정의된 뷰를 사용하여 이러한 테이블에 액세스할 수 있습니다.

데이터 딕셔너리의 특성은 다음과 같습니다.

- 오라클 데이터베이스 서버가 유저, 객체, 제약 조건 및 저장 영역에 대한 정보를 찾는데 사용합니다.
- 객체 구조나 정의가 수정될 때 오라클 데이터베이스에 의해 유지 관리됩니다.
- 유저가 데이터베이스에 대한 정보를 query하는 데 사용합니다.
- SYS 유저가 소유합니다.
- SQL을 사용하여 직접 수정할 수 없습니다.

**주:** DICTIONARY 데이터 딕셔너리 뷰(또는 이에 대한 동의어 DICT)에는 데이터 딕셔너리 테이블 및 뷰의 이름과 설명이 포함됩니다. 뷰 열 및 정의를 보려면 DICT\_COLUMNS 뷰를 사용합니다. 각 뷰의 완전한 정의를 보려면 *Oracle Database Reference*를 참조하십시오. 수백 개의 기본 테이블을 참조하는 1000개가 넘는 뷰가 있습니다.

## 데이터 딕셔너리 뷰

	Query할 수 있는 사람	목차	다음의 하위 집합	참고
DBA_	DBA	모든 항목	N/A	DBA 전용임을 의미하는 추가 열이 있을 수 있습니다.
ALL_	모두	유저가 볼 수 권한을 가진 모든 항목	DBA_ 뷰	유저의 고유 객체와 유저에게 보기 권한이 부여된 기타 객체 포함
USER_	모두	유저가 소유하는 모든 항목	ALL_ 뷰	누락된 OWNER 열을 제외하고는 일반적으로 ALL_과 같음 (일부 뷰에는 PUBLIC 동의어와 같은 약어 이름이 있음)

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 데이터 딕셔너리 뷰

뷰 접두어는 지정된 유저가 볼 수 있는 데이터 및 데이터의 양을 나타냅니다.

모든 항목의 global 뷰는 DBA 권한을 가진 유저만 DBA\_ 접두어를 사용하여 액세스할 수 있습니다.

그 다음 레벨의 권한은 ALL\_ 접두어 레벨이며, 이는 query하는 유저가 소유 여부에 관계없이 볼 수 있는 모든 객체를 나타냅니다. 예를 들어, USER\_A에게 USER\_B가 소유한 테이블에 대한 액세스 권한이 부여된 경우 USER\_A는 테이블 이름을 처리하는 ALL\_ 뷰에 나열된 해당 테이블을 볼 수 있습니다.

USER\_ 접두어는 가장 작은 가시성 범위를 나타냅니다. 이 뷰 유형에는 query하는 유저가 소유한 객체(유저의 고유 스키마에 표시되는 객체)만 표시됩니다.

일반적으로 각 뷰 집합은 행 또는 열 차원에서 더 높은 권한을 가진 뷰 집합의 하위 집합입니다. 주어진 뷰 집합의 모든 뷰가 다른 뷰 집합에 일치하는 뷰를 가지는 것은 아니며 뷰의 정보 특성에 따라 다릅니다. 예를 들어, DBA\_LOCK 뷰는 있지만 ALL\_LOCK 뷰는 없습니다.

그 이유는 DBA만 lock과 관련된 데이터에 관심이 있기 때문입니다. 요구 사항을 충족하는 해당하는 뷰 집합을 선택해야 합니다. DBA 뷰에 대한 액세스 권한을 갖고 있더라도, 결과에 자신이 소유하는 객체에 대한 정보가 표시되고 다른 객체는 결과 집합에 추가되지 않도록 해야 할 경우에는 USER 버전의 뷰만 query해야 할 수 있습니다.

## 데이터 딕셔너리 뷰(계속)

DBA\_ 뷰는 SYSDBA 또는 SELECT ANY DICTIONARY 권한이 있는 유저만 query할 수 있습니다.

모든 딕셔너리 뷰가 DBA\_, ALL\_ 및 USER\_ 접두어로 시작하는 것은 아닙니다. 다음 뷰 또는 뷰에 대한 동의어는 이에 대한 예외입니다.

- AUDIT\_ACTIONS
- CAT
- CHANGE\_PROPAGATIONS
- CHANGE\_PROPAGATION\_SETS
- CHANGE\_SETS
- CHANGE\_SOURCES
- CHANGE\_TABLES
- CLIENT\_RESULT\_CACHE\_STATS\$
- CLU
- COLS
- COLUMN\_PRIVILEGES
- DATABASE\_COMPATIBLE\_LEVEL
- DBMS\_ALERT\_INFO
- DBMS\_LOCK\_ALLOCATED
- DICT
- DICTIONARY
- DICT\_COLUMNS
- DUAL
- GLOBAL\_NAME
- IND
- INDEX\_HISTOGRAM
- INDEX\_STATS
- LOGSTDBY\_UNSUPPORTED\_TABLES
- NLS\_DATABASE\_PARAMETERS
- NLS\_INSTANCE\_PARAMETERS
- NLS\_SESSION\_PARAMETERS
- OBJ
- RECYCLEBIN
- RESOURCE\_COST
- ROLE\_ROLE\_PRIVS
- ROLE\_SYS\_PRIVS
- ROLE\_TAB\_PRIVS
- SEQ
- SESSION\_PRIVS
- SESSION\_ROLES
- SM\$VERSION
- SYN
- TABLE\_PRIVILEGES
- TABS

## 데이터 딕셔너리: 사용 예제

1

```
SELECT table_name, tablespace_name
FROM user_tables;
```

2

```
SELECT sequence_name, min_value, max_value,
increment_by
FROM all_sequences
WHERE sequence_owner IN ('MDSYS', 'XDB');
```

3

```
SELECT USERNAME, ACCOUNT_STATUS
FROM dba_users
WHERE ACCOUNT_STATUS = 'OPEN';
```

4

```
DESCRIBE dba_indexes
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 데이터 딕셔너리: 사용 예제

슬라이드의 예제 query는 다음 질문에 대한 답을 제공합니다.

- 스키마에 생성된 테이블 및 해당 테이블이 있는 테이블스페이스의 이름은 무엇입니까?
- 액세스 권한을 소유한 데이터베이스의 시퀀스에 대한 중요한 정보는 무엇입니까?
- 현재 이 데이터베이스에서 로그인할 수 있는 유저는 누구입니까?
- DBA\_INDEXES 뷰의 열은 무엇입니까? 이 열은 데이터베이스의 모든 인덱스에 대해 볼 수 있는 정보를 보여줍니다. 다음은 이 명령에 대한 출력의 일부입니다.

```
SQL> DESCRIBE dba_indexes
      Name          Null?    Type
----- -----
OWNER           NOT NULL  VARCHAR2(30)
INDEX_NAME      NOT NULL  VARCHAR2(30)
INDEX_TYPE          VARCHAR2(27)
TABLE_OWNER     NOT NULL  VARCHAR2(30)
TABLE_NAME      NOT NULL  VARCHAR2(30)
```

## 퀴즈

Oracle Restart를 사용할 때, 데이터베이스 Instance를 시작 및 정지하려면 SQL\*Plus 대신 Server Control 유ти리티(srvctl)를 사용해야 합니다.

1. 참
2. 거짓

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 퀴즈

데이터베이스의 모든 테이블 이름을 찾을 때 사용할 수 있는 데이터 딕셔너리 뷰는 무엇입니까?

- 1. USER\_TABLES**
- 2. ALL\_TABLES**
- 3. DBA\_TABLES**
- 4. ANY\_TABLES**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 3

## 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 오라클 데이터베이스와 구성 요소 시작 및 정지
- Oracle Enterprise Manager 사용
- SQL\*Plus를 사용하여 데이터베이스 액세스
- 데이터베이스 초기화 파라미터 수정
- 데이터베이스 시작 단계 설명
- 데이터베이스 종료 옵션 설명
- Alert Log 확인
- Dynamic Performance 뷰 액세스

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 4 개요: Oracle Instance 관리

이 연습에서는 다음 내용을 다룹니다.

- Enterprise Manager에서 탐색
- 초기화 파라미터 확인 및 수정
- 데이터베이스 instance 정지 및 시작
- Alert Log 확인
- SQL\*Plus를 사용하여 데이터베이스에 연결

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 5 ASM Instance 관리

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- ASM 사용 시의 이점 설명
- ASM instance 관리
- ASM 디스크 그룹 생성 및 삭제
- ASM 디스크 그룹 확장
- 여러 유ти리티를 사용하여 ASM 메타 데이터 검색

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

이 단원에서는 ASM Instance에 대해 자세히 살펴보고 여러 유ти리티를 사용하여 관리하는 방법에 대해 설명합니다.

## 관리자를 위한 ASM 이점

### ASM에서 불필요한 작업:

- I/O 성능 튜닝
- 데이터 파일 이동 및 재구성
- 파일 이름 관리
- 논리 볼륨 관리
- 파일 시스템 관리
- 클러스터 파일 시스템 관리
- Raw device 관리

### ASM으로 크게 줄어든 업무 부담:

- LUN(Logical Unit Number) 관리
  - 소량화 및 대형화된 LUN
- 시스템 관리자에 대한 데이터베이스 관리자의 의존도
- 수동 유지 관리 작업과 관련된 오류 발생 가능성

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 관리자를 위한 ASM 이점

ASM을 사용하면 비ASM 저장 영역 환경에서 필요한 여러 작업들을 수행하지 않아도 됩니다. 해당 작업들은 다음과 같습니다.

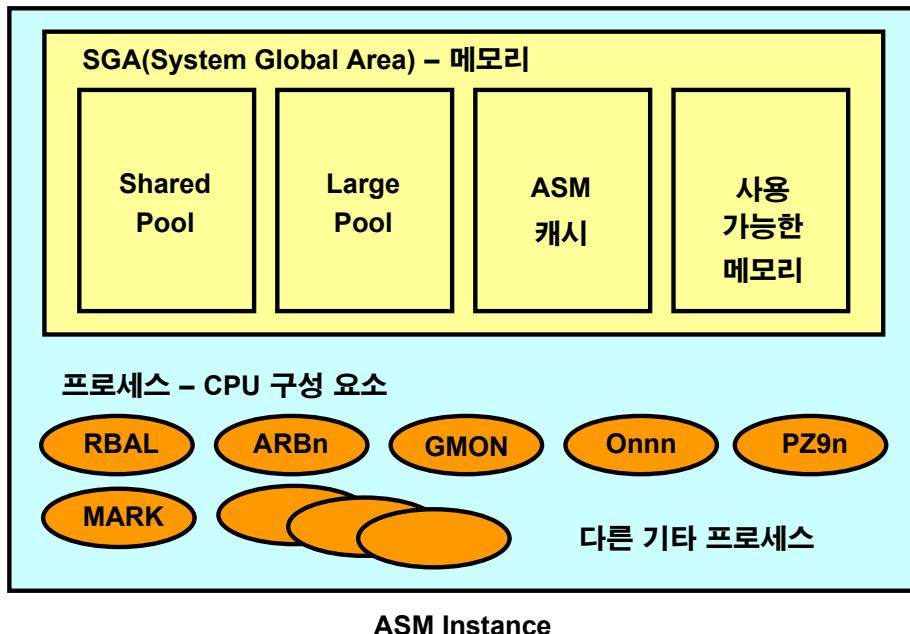
- **I/O 성능 튜닝:** ASM의 stripe-and-mirror-everything 정책과 함께 자동 리밸런싱 작업을 수행할 경우 디스크 활용 및 디스크 핫 스팟을 없애기 위한 I/O 성능 튜닝이 필요하지 않습니다.
- **데이터 파일 이동 및 재구성:** 성능 요구 사항 및 공간 제약을 충족시키기 위해 데이터 파일을 어렵게 배치할 필요가 없습니다.
- **파일 이름 관리:** 더 이상 파일 이름 지정 정책을 정의하고 강제 시행할 필요가 없습니다.
- **논리 볼륨, 파일 시스템, 클러스터 파일 시스템 및 Raw Device 관리:** 더 이상 이러한 저장 영역 요소가 필요하지 않습니다.

ASM은 다음과 같은 중요 영역에서 업무 부담을 줄임으로써 추가적인 이점을 제공합니다.

- ASM에서는 일반적으로 더 적은 수와 더 큰 LUN이 필요하기 때문에 LUN(Logical Unit Number) 관리 부담이 줄었습니다.
- 데이터베이스 관리자 및 시스템 관리자 사이에 종종 존재하던 종속성이 크게 줄었습니다. 예를 들어, 새 데이터 파일을 추가할 때나 한 디스크 그룹에서 다른 디스크 그룹으로 디스크 리소스를 이동할 때 시스템 관리자 작업이 필요하지 않습니다.
- 수동 유지 관리 작업과 관련된 오류 발생 가능성이 크게 줄었습니다. 예를 들어, 기존의 파일 시스템을 사용할 경우 새로 생성된 데이터 파일이 파일 이름 지정 규칙을 실수로 위반하여 남은 데이터베이스에서 백업되지 않을 수도 있습니다.

# ASM Instance

ASM Instance는 ASM을 위한 프로세스 및 메모리 구성 요소의 조합입니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM Instance

ASM 또는 데이터베이스가 시작될 때마다 SGA(System Global Area)라는 공유 메모리 영역이 할당되고 ASM 또는 데이터베이스 백그라운드 프로세스가 시작됩니다. 백그라운드 프로세스와 SGA의 조합을 Oracle ASM Instance 또는 오라클 데이터베이스 Instance라고 부릅니다. Instance는 실행 중인 ASM 환경의 CPU 및 RAM 구성 요소를 나타냅니다.

ASM Instance의 SGA는 데이터베이스 Instance의 SGA와 메모리 할당 및 사용 방법이 다릅니다. ASM Instance의 SGA는 다음과 같이 네 개의 기본 영역으로 구분됩니다.

- **Shared Pool:** 메타 데이터 정보에 사용됨
- **Large Pool:** 병렬 작업에 사용됨
- **ASM 캐시:** 리밸런스 작업 중 읽기 및 쓰기 블록에 사용됨
- **사용 가능한 메모리:** 사용 가능한 할당 해제된 메모리

ASM Instance에 대한 최소 권장 메모리 양은 256MB입니다. 자동 메모리 관리는 ASM Instance에서 기본적으로 활성화되며 개별 SGA 메모리 구성 요소의 크기를 동적으로 튜닝합니다.

ASM Instance에 필요한 메모리 양은 ASM에서 관리 중인 디스크 공간의 양에 따라 달라집니다.

ASM Instance의 두번째 부분은 백그라운드 프로세스입니다. ASM Instance는 여러 백그라운드 프로세스를 포함할 수 있으나, 이러한 프로세스가 항상 모두 제공되는 것은 아닙니다.

## ASM 구성 요소: ASM Instance(계속)

ASM 기능과 관련된 백그라운드 프로세스는 다음 슬라이드에서 다릅니다. 필수 백그라운드 프로세스와 선택적인 백그라운드 프로세스가 있습니다. 이러한 프로세스 중 일부는 다음을 포함할 수 있습니다.

- **ARC<sub>n</sub>**: 아카이버 프로세스
- **CKPT**: 체크포인트 프로세스
- **DBWN**: 데이터베이스 기록자 프로세스
- **DIAG**: 진단 기능 프로세스
- **J<sub>nnn</sub>**: 작업 큐 프로세스
- **LGWR**: 로그 기록자 프로세스
- **PMON**: 프로세스 모니터 프로세스
- **PSP0**: Process spawner 프로세스
- **QMN<sub>n</sub>**: 큐 모니터 프로세스
- **RECO**: 복구자 프로세스
- **SMON**: 시스템 모니터 프로세스
- **VKT<sub>M</sub>**: Virtual keeper of time 프로세스
- **MMAN**: 메모리 관리자 프로세스

위의 프로세스 리스트는 완전한 것이 아닙니다. ASM Instance의 경우 이러한 프로세스는 데이터베이스 Instance에서와 같이 항상 동일한 작업을 수행하지는 않습니다. 예를 들어, 데이터베이스 Instance에서 LGWR 프로세스는 SGA의 로그 버퍼 셱션에서 디스크의 온라인 리두 로그로 변경 벡터를 복사합니다. ASM Instance는 해당 SGA에 로그 버퍼를 포함하거나 온라인 리두 로그를 사용하지 않습니다. ASM Instance의 LGWR 프로세스는 로깅 정보를 ASM 디스크 그룹에 복사합니다.

ASM이 클러스터화된 경우 클러스터 관리와 관련된 추가 프로세스가 ASM Instance에서 실행됩니다. 이러한 일부 프로세스는 다음과 같습니다.

- **LMON**: Global Enqueue Service 모니터 프로세스
- **LMD<sub>n</sub>**: Global Enqueue Service Daemon
- **LMS<sub>n</sub>**: Global Cache Service 프로세스
- **LCK<sub>n</sub>**: Lock 프로세스

## ASM 구성 요소: ASM Instance—기본 프로세스

**ASM Instance 기본 프로세스는 ASM 관련 작업을 수행합니다.**

프로세스	설명
<b>RBAL</b>	검색 중에 모든 장치 파일을 열고 리밸런스 작업을 조정
<b>ARBn</b>	리밸런스 작업을 수행하는 하나 이상의 슬레이브 프로세스
<b>GMON</b>	삭제 또는 오프라인과 같은 디스크 레벨 작업을 관리하고 ASM 디스크 그룹 호환성 개선
<b>MARK</b>	필요한 경우 ASM 할당 단위(AU: Allocation Unit)를 stale로 표시
<b>Onnn</b>	메시지 교환을 위해 ASM Instance에 대한 연결 풀을 형성하는 하나 이상의 ASM 슬레이브 프로세스
<b>PZ9n</b>	클러스터화된 ASM 설치의 데이터를 gv\$ 뷰에서 패치(fetch)하는 데 사용되는 하나 이상의 병렬 슬레이브 프로세스

Copyright © 2010, Oracle. All rights reserved.

### ASM 구성 요소: ASM Instance—기본 프로세스

ASM Instance는 대부분의 기능에 대해 전용 백그라운드 프로세스를 사용합니다. RBAL은 자동 저장 영역 관리(ASM) Instance의 디스크 그룹에 대한 리밸런스 작업을 조정하며, 자동 저장 영역 관리(ASM) 디스크에서 Global 열기를 수행합니다. ARBn 프로세스는 자동 저장 영역 관리(ASM) Instance에서 실제 리밸런스 데이터 Extent 이동을 수행합니다. 한 번에 많은 프로세스가 가능하며 ARB0, ARB1 등으로 이름이 지정됩니다. GMON 프로세스는 ASM 디스크 그룹에서 디스크 멤버를 유지 관리합니다. MARK 프로세스는 오프라인 디스크에 기록을 실패하면 ASM 할당 단위(AU: Allocation Unit)를 stale로 표시합니다. Onnn 프로세스는 클라이언트/서버 연결의 서버측을 나타냅니다. 이러한 프로세스는 Instance가 시작되는 순간에 표시되며 이후에는 사라집니다. 이러한 프로세스는 메시지 교환을 위해 ASM Instance에 대한 연결 풀을 형성하며 필요할 때만 표시됩니다. PZ9n 프로세스는 ASM이 동시에 두 개 이상의 시스템에서 클러스터화된 구성으로 실행될 때 데이터를 패치(fetch)하는 데 사용되는 하나 이상의 병렬 슬레이브 프로세스를 나타냅니다.

# ASM Instance 초기화 파라미터

**ASM Instance는 오라클 데이터베이스 Instance가 사용하는  
파라미터의 작은 부분 집합을 사용합니다.**

```
INSTANCE_TYPE = ASM
ASM_POWER_LIMIT = 1
ASM_DISKSTRING = '/dev/sda1','/dev/sdb*'
ASM_DISKGROUPS = DATA2, FRA
ASM_PREFERRED_READ_FAILURE_GROUPS = DATA.FailGroup2
DIAGNOSTIC_DEST = /u01/app/oracle
LARGE_POOL_SIZE = 12M
REMOTE_LOGIN_PASSWORDFILE = EXCLUSIVE
```

Copyright © 2010, Oracle. All rights reserved.

## ASM Instance 초기화 파라미터

ASM Instance는 일반 데이터베이스 Instance와 같은 방식으로 파라미터 파일을 통해 제어됩니다. 일반적으로 여기서 설정되는 파라미터에는 다음이 포함됩니다.

- INSTANCE\_TYPE은 ASM Instance에 대해 ASM으로 설정되어야 합니다. 이 파라미터만 정의하면 됩니다. 데이터베이스 Instance에 대해서는 값이 RDBMS로 설정됩니다.
- ASM\_POWER\_LIMIT는 리밸런스 작업의 속도를 제어합니다. 값의 범위는 1~11이며, 11이 가장 빠른 속도입니다. 값을 생략하는 경우에는 1이 기본값으로 지정됩니다.
- ASM\_DISKSTRING은 검색에 고려되는 디스크 셋을 제한하기 위해 ASM에서 사용하는 운영 체제 종속 값입니다. 기본값은 널 문자열이며 대부분의 경우에는 널 문자열로도 충분합니다. 위에 표시된 보다 제한적인 값을 사용하면 ASM에서 검색을 수행하는데 필요한 시간이 줄어들 수 있으며, 그 결과 디스크 그룹 마운트 시간이 향상됩니다.
- ASM\_DISKGROUPS는 시작할 때 또는 ALTER DISKGROUP ALL MOUNT 명령을 사용할 때 ASM instance에서 마운트할 디스크 그룹의 이름 리스트입니다. Oracle Restart는 ASM\_DISKGROUPS 파라미터에 나열되지 않았더라도 종속성으로 나열된 경우 디스크 그룹을 마운트할 수 있습니다. 이 파라미터는 기본값이 없습니다.
- ASM\_PREFERRED\_READ\_FAILURE\_GROUPS는 preferred read 디스크를 포함하는 failure 그룹을 지정합니다. 이 그룹은 서버와 매우 근접한 복사본 중 하나로 데이터 복사본을 mirroring한 화장 또는 스트레치 클러스터 데이터베이스에서 유용합니다.

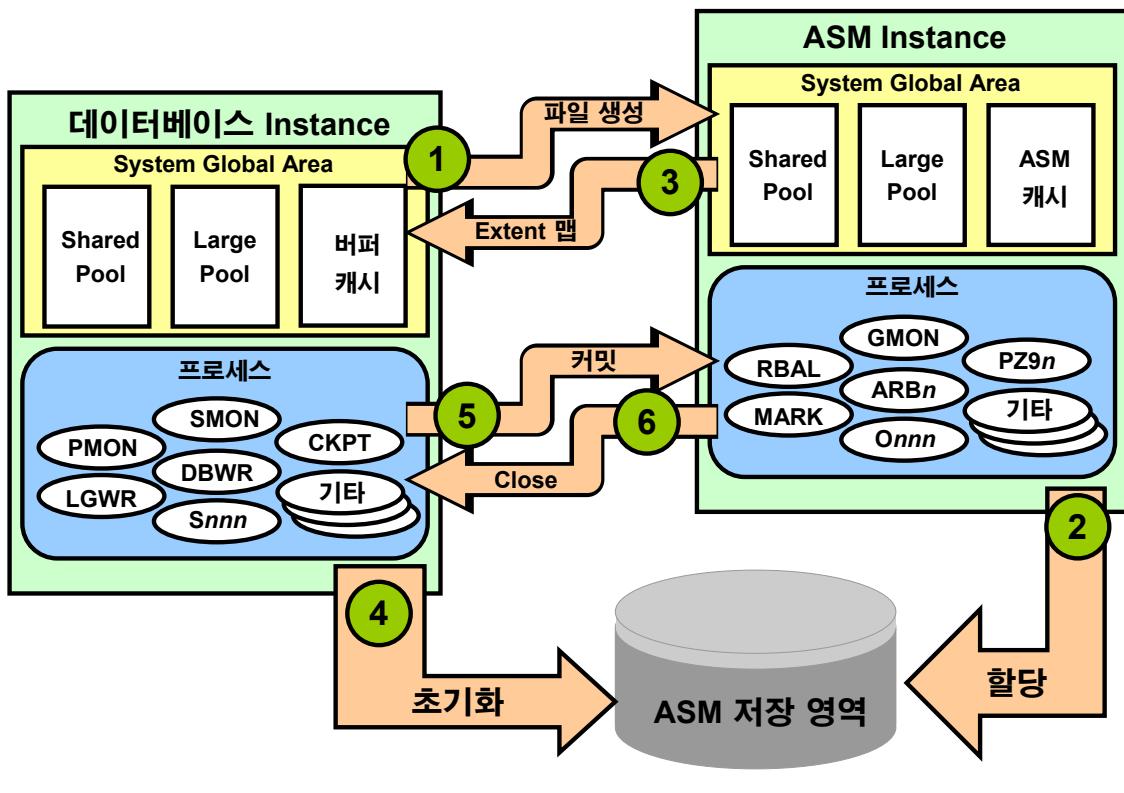
## ASM Instance 초기화 파라미터(계속)

- DIAGNOSTIC\_DEST는 ADR(Automatic Diagnostic Repository) 홈의 위치를 지정합니다. Trace File, Alert Log, 코어 파일 및 Incident 파일은 이 디렉토리 아래에서 찾을 수 있습니다. 이 파라미터의 기본값은 ORACLE\_BASE의 값에서 파생됩니다.
- LARGE\_POOL\_SIZE는 Large pool 할당 힙의 크기를 바이트 단위로 지정합니다. Large pool 할당 힙은 세션 메모리를 위한 Shared Server 시스템, 메시지 버퍼를 위한 병렬 실행 및 디스크 I/O 버퍼를 위한 백업 프로세스에서 사용됩니다. ASM Instance는 자동 메모리 관리 기능을 사용하므로, 이 파라미터는 Large pool을 낮출 수 있는 최소 크기로 사용됩니다.
- REMOTE\_LOGIN\_PASSWORDFILE은 Oracle 소프트웨어가 Password file을 검사할지 여부를 지정합니다. 기본값은 EXCLUSIVE입니다.

위에 나열된 8개 파라미터는 ASM Instance용으로 생성되는 유일한 비기본 파라미터입니다. ASM Instance는 일부 데이터베이스 파라미터가 ASM Instance에 대해 적합하지 않다는 점에서 데이터베이스 Instance와 다릅니다. ASM Instance에서는 전체 344개의 데이터베이스 Instance 파라미터 중 약 74개 정도만 사용할 수 있습니다. 대부분의 설치에서는 기본값만 사용해도 충분하지만 슬라이드에 나열되지 않은 남은 파라미터를 필요에 따라 설정할 수 있습니다.

**주:** MEMORY\_TARGET 파라미터를 명시적으로 설정하지 않더라도 자동 메모리 관리 기능은 ASM Instance에서 기본적으로 활성화됩니다. 이는 전체 ASM 메모리 관리 과정에서 설정해야 하는 유일한 파라미터입니다. 오라클에서는 ASM에 대해 자동 메모리 관리를 사용할 것을 적극 권장합니다.

## 데이터베이스 Instance와 ASM 사이의 상호 작용



Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 Instance와 ASM 사이의 상호 작용

파일 생성 프로세스 그림에서는 데이터베이스 Instance와 ASM 사이에 발생하는 상호 작용을 자세히 보여줍니다. 파일 생성 프로세스는 다음과 같이 수행됩니다.

1. 데이터베이스가 파일 생성을 요청합니다.
2. ASM 포그라운드 프로세스가 COD(Continuing Operation Directory) 항목을 생성하고 디스크 그룹에서 새 파일에 대한 공간을 할당합니다.
3. ASMB 데이터베이스 프로세스가 새 파일에 대한 Extent 맵을 수신합니다.
4. 이제 파일이 열리고 데이터베이스 프로세스가 파일을 직접 초기화합니다.
5. 초기화가 끝나면 데이터베이스 프로세스가 파일 생성을 커밋하도록 요청합니다. 이렇게 하면 ASM 포그라운드 프로세스가 COD 항목을 지우고 파일을 생성된 것으로 표시합니다.
6. 파일 커밋이 확인되면 파일이 암시적으로 닫힙니다. 데이터베이스 Instance는 이후 I/O를 수행할 때 파일을 다시 열어야 합니다.

이 예제에서는 ASM의 구조에 대한 다음 두 가지 중요 사항을 보여줍니다.

- 데이터베이스 Instance와 ASM Instance는 연동된 방식으로 함께 작동합니다. 데이터베이스 Instance는 ASM Extent에 데이터베이스 파일을 매핑하기 위해 ASM과 상호 작용해야 합니다. 또한 데이터베이스 Instance는 ASM Extent를 Lock하거나 이동할 수 있는 ASM 작업 (예: 디스크 그룹 리밸런싱)과 관련한 메시지 스트림을 계속 수신합니다.
- 데이터베이스 I/O는 ASM Instance를 통해 채널링되지 않습니다. 실제로는 슬라이드의 4단계에 설명된 것처럼 데이터베이스가 ASM 파일에 대해 I/O 작업을 직접 수행합니다.

# ASM Instance: Dynamic Performance 뷰

ASM Instance는 Dynamic Performance 뷰로 제공된 메모리 기반 메타 데이터 테이블을 호스트합니다.

- SQL 언어를 사용하여 메타 데이터 전용 정보를 검색하기 위해 ASM 유ти리티에서 액세스됩니다.
- 다음과 같은 여러 ASM 관련 전용 뷰를 포함합니다.

V\$ASM_ALIAS	V\$ASM_ATTRIBUTE	V\$ASM_CLIENT
V\$ASM_DISK	V\$ASM_DISK_IOSTAT	V\$ASM_DISK_STAT
V\$ASM_DISKGROUP	V\$ASM_DISKGROUP_STAT	V\$ASM_FILE
V\$ASM_OPERATION	V\$ASM_TEMPLATE	

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM Instance: Dynamic Performance 뷰

모든 Instance의 주요 기능 중 하나는 메모리 기반 메타 데이터 테이블을 저장하는 기능입니다. 이러한 테이블은 접두어 X\$로 시작하며 일반적으로 기술되어 있지 않습니다. 접두어 V\$로 시작하는 일련의 Dynamic Performance 뷰는 X\$ 메모리 테이블에 포함된 데이터를 커스터마이즈된 방식으로 표시하는 데 사용됩니다. 이 정보는 권한이 있는 관리자만 사용할 수 있는 읽기 전용 방식으로 제공됩니다. 이 정보는 SQL 언어를 사용하여 ASM에서 검색됩니다. 위 슬라이드에는 ASM 관련 메타 데이터를 포함하는 가장 일반적인 Dynamic Performance 뷰가 나열되어 있습니다. 이 외에도 Dynamic Performance 뷰가 수백 개 더 있지만 데이터베이스 Instance가 데이터베이스 콘트롤 파일을 마운트해야 하기 때문에 이러한 대부분의 뷰는 비어 있습니다. ASM Instance는 데이터베이스 콘트롤 파일을 마운트하지 않습니다. Dynamic Performance 뷰의 전체 리스트를 보려면 *Oracle Database Reference 11g Release 2(11.2)* 설명서를 참조하십시오.

# ASM 시스템 권한

- ASM Instance에는 데이터 딕셔너리가 없으므로 ASM에 연결하려면 다음 시스템 권한을 사용할 수 밖에 없습니다.

ASM 권한	권한 그룹 (권장)	권한
SYSASM	OSASM (asmadmin)	모든 관리 권한
SYSDBA	OSDBA (asmdba)	ASM에 저장된 데이터에 대한 액세스 권한 및 현재 릴리스의 SYSASM
SYSOPER	OSOPER (asmoper)	비파괴적인 ALTER DISKGROUP 명령과 함께 ASM Instance를 시작 및 정지할 수 있는 제한된 권한

- SYS 유저는 SYSASM 권한으로 자동 생성됩니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM 시스템 권한

ASM Instance에는 데이터 딕셔너리가 없으므로 ASM Instance에 연결하기 위해서는 SYSASM, SYSDBA 또는 SYSOPER의 세 가지 시스템 권한 중 하나를 사용할 수 밖에 없습니다.

다음 리스트에서는 이러한 ASM 시스템 권한에 대해 소개합니다.

- **SYSASM:** 이 권한은 ASM Instance에 대한 모든 관리 권한을 제공합니다.
- **SYSDBA:** 이 권한은 ASM에 저장된 데이터에 대한 액세스 권한을 부여하고, 현재 릴리스에서 SYSASM 관리 권한을 부여합니다.
- **SYSOPER:** 이 권한은 비파괴적인 일련의 ALTER DISKGROUP 명령과 함께 ASM Instance를 시작 및 정지할 수 있는 기능을 부여합니다. CREATE DISKGROUP과 같은 다른 명령은 허용되지 않습니다.

ASM이 설치된 경우 SYSASM, SYSDBA 및 SYSOPER 권한을 인증하기 위해 운영 체제 그룹이 사용됩니다. 이러한 그룹은 OUI(Oracle Universal Installer) 유ти리티에서 각각 OSASM, OSDBA 및 OSOPER 그룹으로 참조됩니다. 즉, OSASM 그룹은 SYSASM 권한이 부여된 운영 체제 그룹입니다. OSASM, OSDBA 및 OSOPER 그룹 값에 대해 권장되는 이름은 각각 asmadmin, asmdba 및 asmoper입니다. 따라서 SYSASM은 데이터베이스에서, OSASM은 OUI 유ти리티에서, asmadmin은 운영 체제에서 사용되는 이름입니다. 모두 동일한 유저 그룹을 참조합니다. ASM Instance를 처음 생성하면 ASM 유저로 sys 및 asmsnmp만 정의됩니다.

# Enterprise Manager를 사용하여 ASM 유저 관리

**Edit User: SYS**

To allow users to connect to the ASM instance through remote connection using password file authentication, the user needs to be created and granted with privileges. The password file has to be created using the ORAPWD utility already and the REMOTE\_LOGIN\_PASSWORDFILE initialization parameter needs to be set to EXCLUSIVE. In a cluster environment, creating or editing a user on one node creates or edits that user for all other running nodes of the cluster automatically.

**Login Credential**

User Name: SYS  
Password:   
Confirm Password:

**Create User**

To allow users to connect to the ASM instance through remote connection using password file authentication, the user needs to be created and granted with privileges. The password file has to be created using the ORAPWD utility already and the REMOTE\_LOGIN\_PASSWORDFILE initialization parameter needs to be set to EXCLUSIVE. In a cluster environment, creating or editing a user on one node creates or edits that user for all other running nodes of the cluster automatically.

**Login Credential**

\* User Name: MFULLER  
\* Password:  \* Confirm Password:

**Privileges**

**Available Privileges**: SYSDBA, SYSOPER

**Granted Privileges**: SYSASM

> Move, >> Move All, < Remove, << Remove All

## Enterprise Manager를 사용하여 ASM 유저 관리

Enterprise Manager에서는 원격 연결을 통해(Password file 인증 사용) ASM Instance에 액세스하는 유저를 관리할 수 있습니다. 이러한 유저는 해당 ASM Instance에 대해서만 예약됩니다.

이 기능은 SYSASM 유저로 연결된 경우에만 사용할 수 있으며, SYSDBA 또는 SYSOPER 유저로 연결하면 표시되지 않습니다.

- Create 버튼을 누르면 Create User 페이지가 표시됩니다.
- Edit 버튼을 누르면 Edit User 페이지가 표시됩니다.
- Delete 버튼을 누르면 생성된 유저를 삭제할 수 있습니다.

**주:** SYSASM 를로 ASM에 로그인하려면 페이지 최상단에 있는 Preferences 링크를 누른 다음 Preferred Credentials 링크를 누릅니다. 그러면 대상 유형의 리스트에 ASM이 표시됩니다. ASM 대상 유형 옆에 있는 Set Credentials 아이콘을 눌러 SYSASM 권한이 있는 계정 및 암호를 정의합니다. 변경 내용을 적용하려면 Database Control에서 로그아웃한 다음 다시 로그인해야 합니다.

# SQL\*Plus를 사용하여 ASM Instance 시작 및 정지

**SQL\*Plus를 사용하여 ASM Instance를 시작 및 정지하는 방법은 데이터베이스 Instance를 시작 및 정지하는 방법과 비슷합니다.**

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ sqlplus / AS SYSASM
SQL*Plus: Release 11.2.0.1.0 - Production on Wed Jul 8 20:46:46 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to an idle instance.
SQL> startup
ASM instance started

Total System Global Area  284565504 bytes
Fixed Size                  1336028 bytes
Variable Size                258063652 bytes
ASM Cache                   25165824 bytes
ASM diskgroups mounted
ASM diskgroups volume enabled
SQL> shutdown abort
```

Copyright © 2010, Oracle. All rights reserved.

## SQL\*Plus를 사용하여 ASM Instance 시작 및 정지

SQL\*Plus에서는 오라클 데이터베이스 Instance를 시작할 때와 비슷한 방법으로 STARTUP 명령을 사용하여 ASM Instance를 시작합니다. ASM Instance를 시작할 때는 다음 사항을 주의하십시오.

- SQL\*Plus로 ASM Instance에 연결하려면 ORACLE\_SID 환경 변수를 ASM SID로 설정합니다. 단일 Instance 데이터베이스에 대한 기본 ASM SID는 +ASM이고, Oracle RAC 노드용 ASM에 대한 기본 SID는 +ASMnode\_number입니다. 여기서 *node\_number*는 노드의 번호입니다. oraenv 스크립트는 ORACLE\_BASE, ORACLE\_SID, ORACLE\_HOME 및 PATH 변수를 설정합니다.
- 초기화 파라미터 파일에는 다음 항목이 포함되어야 합니다.

```
INSTANCE_TYPE = ASM
```

이 파라미터는 데이터베이스 Instance가 아닌 ASM Instance가 시작 중임을 나타냅니다.

- 데이터베이스를 마운트하고 열려는 것이 아니라 STARTUP 명령을 실행하면 초기화 파라미터 ASM\_DISKGROUPS에서 지정된 디스크 그룹을 마운트하려고 시도합니다. ASM\_DISKGROUPS에 대한 값을 입력하지 않은 경우 나중에 ALTER DISKGROUP...MOUNT 명령으로 디스크 그룹을 마운트할 수 있습니다.

## SQL\*Plus를 사용하여 ASM Instance 시작 및 정지(계속)

다음 리스트에서는 ASM과 관련된 STARTUP 명령 파라미터를 설명합니다.

- **FORCE:** 재시작 전에 ASM Instance에 대해 SHUTDOWN ABORT를 실행합니다.
- **MOUNT 또는 OPEN:** ASM\_DISKGROUPS 초기화 파라미터에 지정된 디스크 그룹을 마운트합니다. 명령 파라미터가 지정되지 않은 경우 기본값입니다.
- **NOMOUNT:** 디스크 그룹을 마운트하지 않고 ASM Instance를 시작합니다.
- **RESTRICT:** Instance를 제한 모드로 시작합니다. RESTRICT 절은 MOUNT, NOMOUNT 및 OPEN 절과 함께 사용할 수 있습니다.

제한 모드에서 데이터베이스 Instance는 디스크 그룹을 사용할 수 없습니다. 즉, 데이터베이스가 해당 디스크 그룹에 있는 파일을 열 수 없습니다. 또한 디스크 그룹이 제한 모드에 있는 Instance에 의해 마운트된 경우 해당 디스크 그룹을 클러스터에 있는 다른 Instance에서 마운트할 수 없습니다. 제한 모드에서는 클라이언트의 간섭 없이 디스크 그룹에서 유지 관리 작업을 수행할 수 있습니다. 디스크 그룹이 제한 모드에 있는 동안 리밸런스 작업을 수행하면 클러스터 환경의 ASM Instance 사이에 Extent 맵 Lock 및 Lock 해제 메시징을 수행할 필요가 없습니다. 따라서 전반적인 리밸런스 처리 능력이 향상됩니다. 유지 관리 기간이 끝나면 디스크 그룹을 명시적으로 마운트 해제하고 일반 모드로 다시 마운트해야 합니다.

ASM 종료 프로세스는 유저가 SQL\*Plus에서 SHUTDOWN 명령을 실행할 때 시작됩니다. 이 명령을 실행하기 전에 ASM Instance에 연결할 수 있도록 ORACLE\_SID 및 ORACLE\_HOME 환경 변수가 설정되었는지 확인하십시오.

오라클은 ASM Instance를 종료하려고 시도하기 전에 ASM Instance를 사용하는 모든 데이터베이스 Instance를 종료할 것을 적극 권장합니다.

다음 리스트에서는 ASM과 관련된 SHUTDOWN 명령 파라미터를 설명합니다.

- **NORMAL:** 모든 디스크 그룹을 마운트 해제하고 ASM Instance를 종료하기 전에 모든 진행 중인 SQL이 완료될 때까지 기다립니다. Instance가 종료되기 전에 현재 연결되어 있는 모든 유저가 Instance에서 연결을 해제할 때까지 기다립니다. ASM Instance에 연결되어 있는 데이터베이스 Instance가 있으면 SHUTDOWN 명령이 오류를 반환하고 ASM Instance는 계속 실행 상태로 유지됩니다. NORMAL이 기본 종료 모드입니다.
- **IMMEDIATE 또는 TRANSACTIONAL:** 모든 디스크 그룹을 마운트 해제하고 ASM Instance를 종료하기 전에 모든 진행 중인 SQL이 완료될 때까지 기다립니다. ASM은 현재 Instance에 연결된 유저의 연결이 해제될 때까지 기다리지 않습니다. ASM Instance에 연결되어 있는 데이터베이스 Instance가 있으면 SHUTDOWN 명령이 오류를 반환하고 ASM Instance는 계속 실행 상태로 유지됩니다.
- **ABORT:** 디스크 그룹을 순차적으로 마운트 해제하지 않고 ASM Instance가 즉시 종료됩니다. 따라서 다음에 ASM을 시작하면 Recovery가 발생합니다. ASM Instance에 연결된 데이터베이스 Instance가 있으면 데이터베이스 Instance가 중지됩니다.

**주:** NORMAL, IMMEDIATE 및 TRANSACTIONAL의 종료 형식은 연결된 RDBMS Instance가 있는 경우 적용되지 않습니다. 다음 오류가 반환됩니다.

ORA-15097: cannot SHUTDOWN ASM instance with connect  
RDBMS instance

## srvctl을 사용하여 ASM Instance 시작 및 정지

Server Control 유틸리티(srvctl)를 사용하여 ASM Instance를 시작 및 정지할 수 있습니다.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ srvctl start asm -o mount
$ srvctl stop asm -f
```

Sever Control 유틸리티(srvctl)를 사용하여 ASM Instance의 상태를 확인할 수 있습니다.

```
$ srvctl status asm
ASM is running on edrsr25p1
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### srvctl을 사용하여 ASM Instance 시작 및 정지

Server Control 유틸리티(srvctl)를 사용하여 그리드 Infrastructure에 의해 관리되는 다른 리소스와 함께 ASM Instance를 시작 및 정지할 수 있습니다. srvctl 유틸리티는 그리드 Infrastructure에 대한 ORACLE\_HOME/bin 위치 및 데이터베이스 설치의 ORACLE\_HOME/bin 위치에서 찾을 수 있습니다. ASM, 리스너 또는 Oracle Restart를 관리할 때 그리드 Infrastructure ORACLE\_HOME에 있는 srvctl 유틸리티를 사용할 수 있습니다. srvctl 유틸리티를 사용하면 다음과 같은 방식으로 ASM을 제어할 수 있습니다.

- ASM Instance를 시작합니다.

```
srvctl start asm [-o <start_option>]
<start_option>은 적합한 Instance 시작 옵션 중 하나입니다
(FORCE, MOUNT, OPEN, NOMOUNT 또는 RESTRICT)(선택 사항).
```

- ASM Instance 정지.

```
srvctl stop asm [-o <stop_option>] -f
<stop_option>은 적합한 Instance 종료 옵션 중 하나이며
(NORMAL, IMMEDIATE, TRANSACTIONAL 또는 ABORT)(선택 사항),
-f는 강제 실행 스위치입니다.
```

- ASM Instance의 상태를 보고합니다.

```
srvctl status asm
```

## asmcmd를 사용하여 ASM Instance 시작 및 정지

**asmcmd 유ти리티는 SQL 언어를 사용하지 않고 ASM에 명령행 인터페이스를 제공합니다.**

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ asmcmd
Connected to an idle instance.
ASMCMD> startup
ASM instance started

Total System Global Area  284565504 bytes
Fixed Size                  1336028 bytes
Variable Size                258063652 bytes
ASM Cache                   25165824 bytes
ASM diskgroups mounted
ASM diskgroups volume enabled
ASMCMD> shutdown --abort
ASM instance shut down
Connected to an idle instance.
```

Copyright © 2010, Oracle. All rights reserved.

### asmcmd를 사용하여 ASM Instance 시작 및 정지

관리를 위한 ASM 메타 데이터는 ASM Instance 내에 포함된 Dynamic Performance 뷰 안에 있습니다. 이러한 뷰는 일반적으로 SQL 언어를 사용하여 ASM 유ти리티에서 액세스됩니다. SQL 언어 지식을 배우려면 ASM을 학습하는 기간이 들어날 수 있으며, SQL 언어 교육은 일반적으로 시스템 관리자 또는 저장 영역 관리자의 학습 과정에서 필수 요건이 아닙니다. asmcmd 유ти리티는 일반 ASM 관리 작업을 위해 UNIX 스타일의 구문을 사용할 수 있는 pseudo 셸과 같은 환경을 제공합니다. 이 유ти리티를 사용하면 Oracle ASM instance, 디스크 그룹, 디스크 그룹을 위한 파일 액세스 제어, 디스크 그룹 내부의 파일 및 디렉토리, 디스크 그룹에 대한 템플리트 및 볼륨을 관리할 수 있습니다.

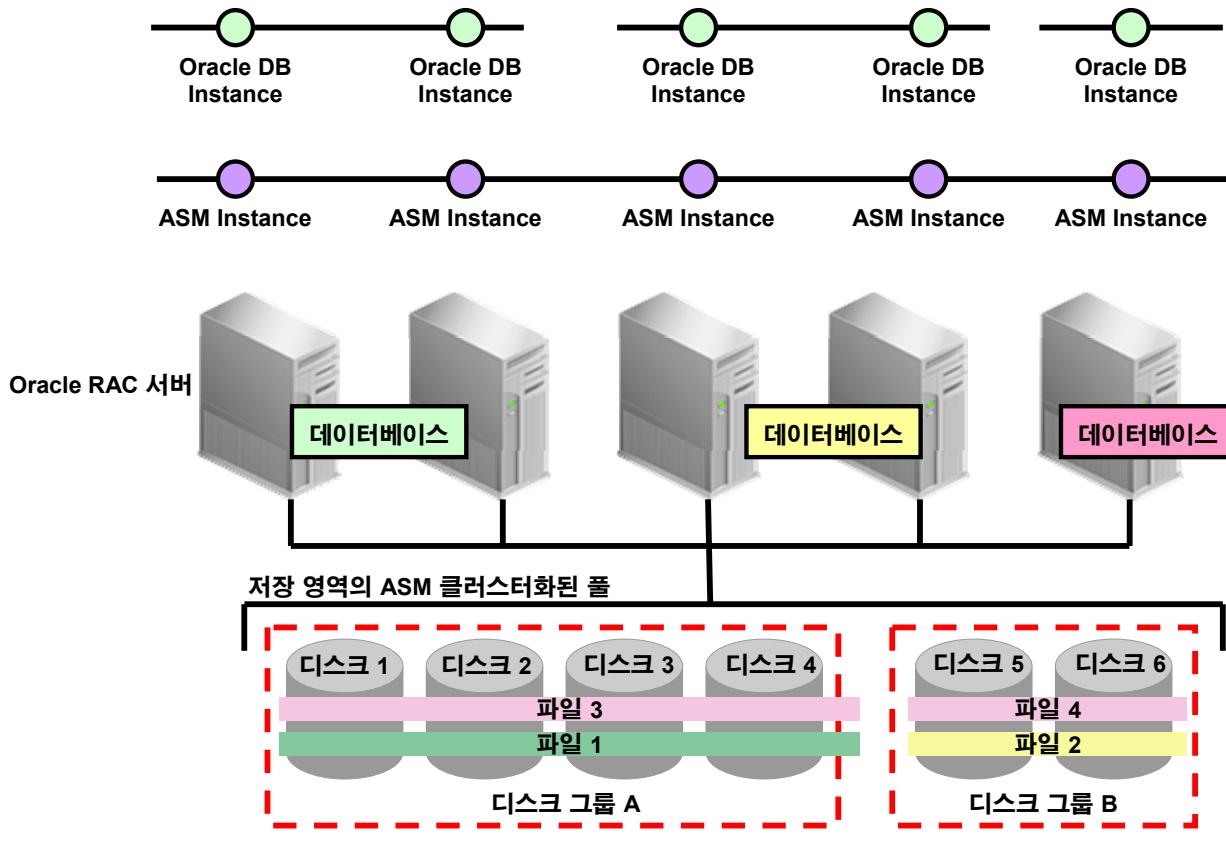
asmcmd 유ти리티를 사용하여 ASM Instance를 시작 및 종료할 수 있습니다. 지원되는 시작 옵션은 다음과 같습니다.

- nomount (디스크 그룹을 마운트하지 않고 ASM Instance 시작)
- restrict (ASM Instance 시작 및 데이터베이스 사용 제한)
- pfile <pfile.ora> (커스텀 pfile로 ASM Instance 시작)

지원되는 종료 옵션은 다음과 같습니다.

- immediate (즉시 종료 수행)
- abort (기존의 모든 작업 중지)

## 디스크 그룹 개요



Copyright © 2010, Oracle. All rights reserved.

**ORACLE**

### 디스크 그룹 개요

디스크 그룹은 ASM이 하나의 모음으로 관리하는 하나 이상의 디스크에 대한 논리적 그룹입니다. 각 디스크 그룹에는 그 자체와 연관된 메타 데이터가 포함됩니다. ASM 디스크 그룹은 일반적인 저장 영역 네트워크의 논리 볼륨과 비슷한 개념으로 이해할 수 있습니다.

파일은 디스크 그룹 내의 공간에서 할당됩니다. 디스크 그룹에 저장된 파일의 내용은 핫 스팟을 없애고 디스크 간 동일 성능을 제공하기 위해 디스크 그룹 내의 디스크에 고르게 분산 또는 스트라이핑됩니다. 각 ASM 파일은 단일 디스크 그룹 내에 완전히 포함됩니다. 그러나 디스크 그룹에는 여러 데이터베이스에 속하는 파일이 포함될 수 있으며, 단일 데이터베이스는 여러 디스크 그룹의 서로 다른 파일을 사용할 수 있습니다.

디스크 그룹의 주요 속성은 중복성 설정입니다. 디스크 그룹에서 가능한 중복성 설정은 세 가지입니다.

- External redundancy - ASM이 mirroring을 제공하지 않으며, 디스크가 매우 안정적인 것으로 간주됩니다.
- Normal redundancy - ASM이 기본적으로 양방향 mirroring을 지원하여 안정성이 낮은 저장 영역에 대한 데이터 무결성을 보장합니다.
- High redundancy - ASM이 기본적으로 세 방향 mirroring을 지원하여 더 높은 레벨의 데이터 무결성을 보장합니다.

ASM에서는 최대 63개까지 디스크 그룹 생성을 지원하지만 대부분의 설치에서는 디스크 그룹을 몇 개 이상으로 사용하는 경우가 매우 드뭅니다.

# ASM 디스크

## ASM 디스크:

- **ASM 디스크 그룹에 프로비전된 저장 장치**
- **일반 O/S 인터페이스를 통해 액세스됨**
- **ASM 소유자가 읽기 및 쓰기 액세스할 수 있어야 함**
- **클러스터의 모든 노드가 액세스할 수 있어야 함**
- **각 노드에서 O/S 이름 또는 경로가 서로 달라야 함**
- **가능한 형태:**
  - 전체 물리적 디스크 또는 물리적 디스크의 partition
  - 저장 영역 배열의 디스크 또는 partition
  - LV(논리 블루) 또는 LUN(논리 단위)
  - NFS(네트워크 연결 파일)

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## ASM 디스크

ASM 디스크 그룹은 하나 이상의 ASM 디스크를 포함합니다.

ASM 디스크는 클러스터의 모든 노드에서 ASM 소유자가 읽기 및 쓰기 액세스할 수 있어야 합니다.

ASM 디스크는 디스크에서 하나의 partition일 수 있지만 Oracle은 ASM에서 사용되는 물리적 디스크를 다른 응용 프로그램과 공유하지 않을 것을 적극 권장합니다. 디스크 그룹 내에서 ASM 디스크의 I/O 성능은 ASM 스트라이핑 및 Mirror 정책이 가장 잘 적용될 수 있도록 비슷하고 일관적이어야 하기 때문입니다. ASM과 다른 응용 프로그램 사이에 장치를 공유하면 비슷하고 일관적인 디스크 성능을 보장하기가 어렵습니다.

클러스터의 개별 노드에 있는 ASM 디스크의 O/S 장치 이름은 동일할 필요가 없습니다.

ASM은 ASM 디스크의 헤더를 읽어 디스크 그룹의 멤버를 식별합니다.

일반적으로 ASM 디스크는 ASM에 제공되는 저장 영역 배열의 Raw LUN입니다. 또한 ASM 디스크는 원격 NFS 파일리에 있는 파일일 수도 있습니다.

## 할당 단위(AU: Allocation Unit)

ASM 디스크는 할당 단위(AU: Allocation Unit)로 나뉩니다.

- AU 크기는 디스크 그룹 생성 시 구성할 수 있습니다.
- 기본 AU 크기는 1MB입니다.
  - 이 크기는 데이터베이스가 캐시하기에 충분히 작은 크기이며, 효율적인 순차적 액세스를 위해서도 충분히 큰 크기입니다.
- 허용 가능한 AU 크기:
  - 1, 2, 4, 8, 16, 32 또는 64MB
  - AU가 크면 VLDB(very large database) 시나리오 또는 특별한 저장 하드웨어를 사용할 때 매우 유용할 수 있습니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 할당 단위(AU: Allocation Unit)

ASM 디스크 내에서 공간은 할당 단위(AU: Allocation Unit)로 나뉩니다. 기본 AU 크기는 1MB이며, 이 크기는 핫 스팟이 발생하지 않도록 하기에 충분히 작은 크기이지만, 효율적인 순차적 액세스를 제공하기에도 충분히 큰 크기입니다. AU 크기는 디스크 그룹을 생성할 때 설정할 수 있습니다. 디스크 그룹의 AU 크기는 변경할 수 없습니다. AU 크기가 크면 VLDB(very large database) 시나리오 또는 특별한 저장 하드웨어를 사용할 때 매우 유용할 수 있습니다. AU가 지속적으로 액세스되는 경우 기본 커널이 추가 액세스 효율을 높이기 위해 이를 캐시에 저장합니다.

# ASM 파일

## ASM 파일:

- AU로 구성되는 ASM Extent의 모음
  - 가변 크기 Extent로 큰 파일 지원
- 데이터베이스 커널에는 일반 파일로 표시됨
- 파일 이름이 '+'로 시작
  - 예: +DATA/orcl/datafile/system.256.689832921
- 선택적인 alias 파일 이름과 연관될 수 있음
  - 예: +DATA/dbfiles/mydb/system01.dbf
- 디스크 그룹의 여러 디스크에 고르게 분산됨
- 디스크 그룹에 정의된 정책에 따라 Mirroring됨

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## ASM 파일

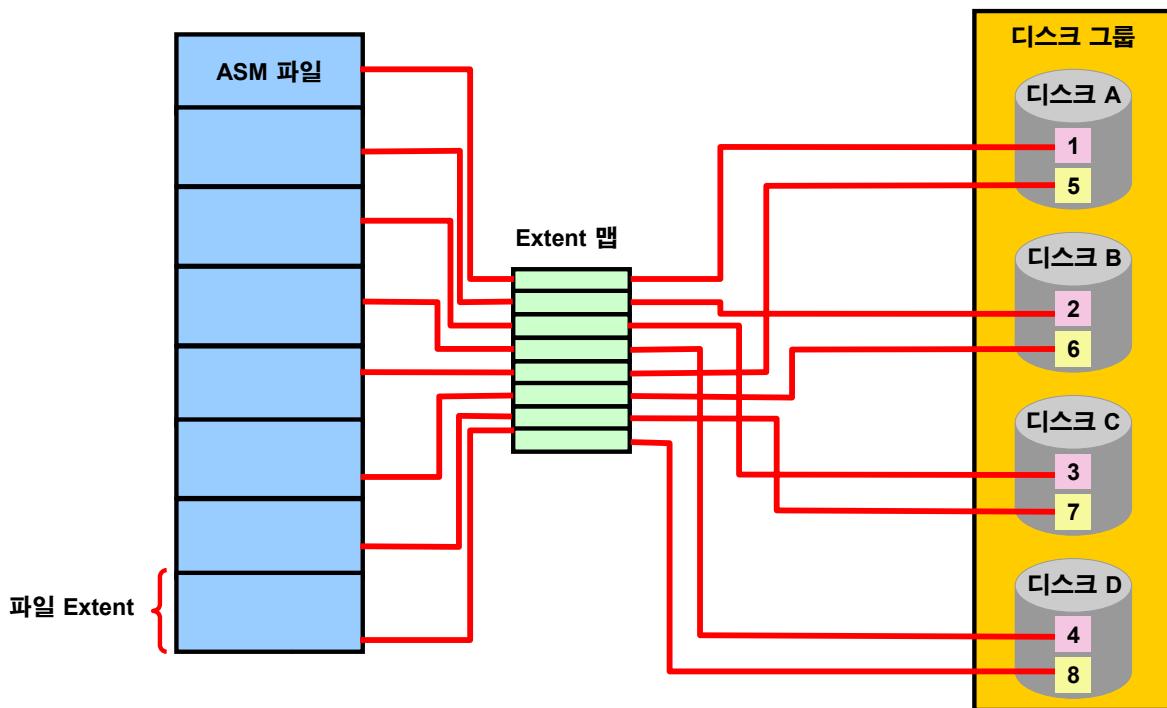
ASM은 ASM의 클라이언트가 사용할 수 있는 파일 집합을 나타냅니다. ASM 파일은 할당 단위 (AU: Allocation Unit) 집합으로 구성되며 데이터베이스 커널에 일반 파일로 표시됩니다.

각 ASM 파일에는 고유한 시스템 생성 이름이 포함됩니다. 슬라이드에서는 전체 ASM 파일 이름의 예를 보여줍니다. 전체 ASM 파일 이름은 디스크 그룹 이름에 더하기 기호로 시작하는 계층으로 표시됩니다. 디스크 그룹 이름 앞에는 데이터베이스 이름과 파일 유형이 옵니다. 마지막 요소는 태그 이름, 파일 번호 및 incarnation 번호로 구성됩니다. 선택적으로 관리자가 ASM 파일을 보다 효율적으로 참조할 수 있도록 alias를 생성할 수도 있습니다.

파일은 SAME(stripe and mirror everything) 정책을 사용하여 디스크 그룹의 ASM 디스크에 고르게 분산됩니다.

ASM은 기본적으로 데이터 파일, 로그 파일, 콘트롤 파일, RMAN 백업 등과 같은 대부분의 데이터베이스 관련 파일 유형을 지원합니다. Oracle Database 11g Release 2 이전까지 ASM에서는 오라클 데이터베이스 관련 파일만 지원되었으며, ASM을 사용하여 ASCII trace file 및 alert log, Oracle 이진 파일, OCR(Oracle Cluster Registry) 및 클러스터 voting disk를 저장 및 관리할 수 없었습니다. Oracle Database 11g Release 2에서는 ASM에서 범용 파일 시스템을 실행할 수 있는 방법을 제공하여 이러한 제한을 없앴습니다.

## Extent 맵



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Extent 맵

ASM은 Extent 맵이라고 부르는 메타 데이터로 파일 레이아웃을 추적합니다. Extent 맵은 파일에 있는 데이터 Extent를 디스크의 할당 단위(AU: Allocation Unit)에 매핑하는 테이블입니다.

파일 Extent와 할당 단위 사이의 관계는 다음과 같습니다. Extent에는 다음과 같은 내용이 포함됩니다.

- 처음 20,000개의 Extent(0–19999)대한 1개의 AU
- 다음 20,000개의 Extent(20000–39999)에 대한 4개의 AU
- 40,000개 이상의 Extent에 대한 16개의 AU

큰 할당 단위와 함께 가변 크기 Extent를 사용하여 매우 큰 ASM 파일을 수용할 수 있습니다.

## 스트라이핑 세분성

**ASM는 로드 밸런싱 및 대기 시간에 대한 스트라이핑을 구분해서 사용:**

- **상위단계(coarse-grain) 스트라이핑은 로드 밸런싱을 위해 할당 단위(AU: Allocation Unit)를 연결합니다.**
  - 예:



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 스트라이핑 세분성

ASM에서 스트라이핑은 다음과 같은 두 가지 주요 목적을 갖습니다.

- 디스크 그룹 내의 모든 디스크에서 I/O 로드 밸런싱
- I/O 대기 시간 개선

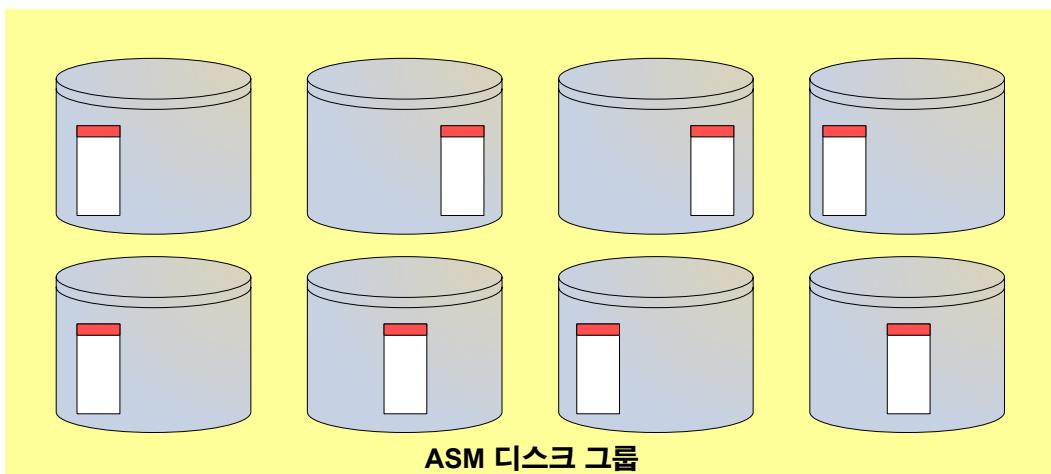
상위단계(coarse-grain) 스트라이핑은 디스크 그룹 내의 디스크에서 할당 단위(AU: Allocation Unit)를 분산합니다. 이를 통해 디스크 그룹에 대한 로드 밸런싱을 제공합니다. 파일이 할당되면 ASM이 모든 디스크에서 할당 단위를 고르게 분산합니다. 일부 경우에는 분산이 완전히 고르지 않을 수 있지만 시간이 지날수록 거의 동일하게 분산됩니다. 위 다이어그램에서는 총 8개의 디스크를 포함하는 external redundancy 디스크 그룹에서 5개의 디스크에 5개 할당 단위가 스트라이핑된 파일을 보여줍니다.

처음 20,000 Extent에서 Extent 크기는 AU 크기와 동일합니다. 20,000개를 지나서 최대 40,000 개까지의 Extent에서는 Extent 집합이 항상 한 번에 8개 할당되고 Extent 크기는 4\*AU 크기와 동일합니다. AU 크기가 1MB인 경우, ASM 파일은 한 번에 64MB가 증가합니다( $8 * 4 * 1 \text{ MB}$ ). 파일이 상위단계(coarse-grain) 스트라이핑된 경우 1 AU의 스트라이프로 8개 Extent 집합에 걸쳐 스트라이핑됩니다. 스트라이핑은 항상 Extent 레벨이 아닌 AU 레벨에서 수행됩니다. 따라서 상위단계(coarse-grain) 파일의 모든 AU는 파일 크기에 관계없이 해당 파일의 이전 AU와 다른 디스크에 있습니다. 40,000개 이후의 Extent에서는 Extent가 계속 한 번에 8개 할당되지만 Extent 크기는 16\*AU 크기와 동일합니다.

## 하위단계(Fine-Grain) 스트라이핑

하위단계(Fine-grain) 스트라이핑은 대기 시간 개선을 위해 할당 단위(AU: Allocation Unit) 그룹에 128KB 스트라이핑 단위를 사용합니다.

- 8개 디스크 및 external redundancy를 포함하는 디스크 그룹
- 사용 중인 기본 AU 크기(1MB)
- 처음 1MB extent는 8개 AU에서 128KB 스트라이프로 기록됨



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 하위단계(Fine-Grain) 스트라이핑

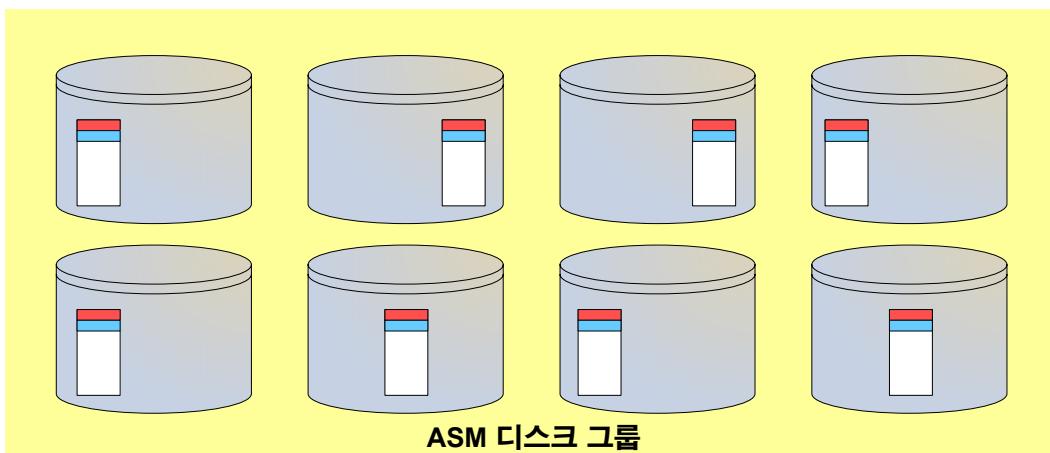
하위단계(Fine-grain) 스트라이핑은 데이터 Extent를 128KB 조각으로 분할합니다. 이렇게 하는 이유는 각 Extent에 대한 로드를 여러 디스크로 분산하여 특정 파일 유형에 대한 대기 시간을 개선하기 위한 것입니다. 하위 단계 스트라이핑은 기본적으로 콘트롤 파일 및 온라인 리두 로그 파일에 사용됩니다.

이 페이지의 다이어그램에서는 하위단계 스트라이핑의 작동 방식을 보여줍니다. 이 예제에서, 새 파일의 처음 1MB Extent는 디스크 그룹에 있는 8개 디스크에 분산된 8개의 서로 다른 할당 단위(AU: Allocation Unit) 중 처음 128KB를 점유합니다. 따라서 1MB의 읽기 또는 쓰기가 디스크 1개가 아닌 8개 디스크에 분산됩니다.

## 하위단계(Fine-Grain) 스트라이핑

예제:

- 8개 디스크 및 external redundancy를 포함하는 디스크 그룹
- 사용 중인 기본 AU 크기(1MB)
- 다음 1MB Extent는 할당 단위가 전부 채워질 때까지 동일한 8개 할당 단위(AU: Allocation Unit)에 분산되는 128KB 스트라이프로 기록됨



ORACLE®

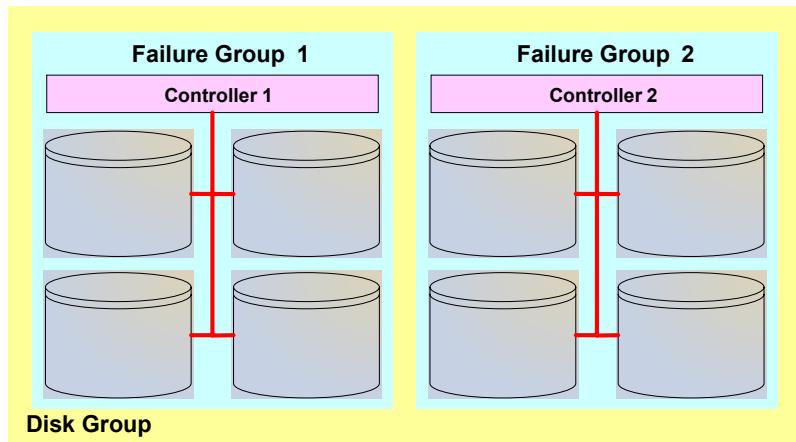
Copyright © 2010, Oracle. All rights reserved.

### 하위단계(Fine-Grain) 스트라이핑(계속)

이전 예제에서 계속하여, 다음 1MB Extent 공간은 동일한 각 할당 단위(AU: Allocation Unit)의 두번째 128KB에 분산됩니다. 이 패턴은 처음 할당 단위 집합이 채워지고 다른 집합이 할당될 때까지 계속됩니다.

# ASM Failure 그룹

- Failure 허용이 필요한 공통 리소스를 공유하는 디스크 모음
- 별개의 Failure 그룹에 저장된 Mirroring된 Extent 복사본
- 저장 하드웨어로 Failure 그룹 경계 지정
  - 디스크 컨트롤러 분리 기반 예제:



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM Failure 그룹

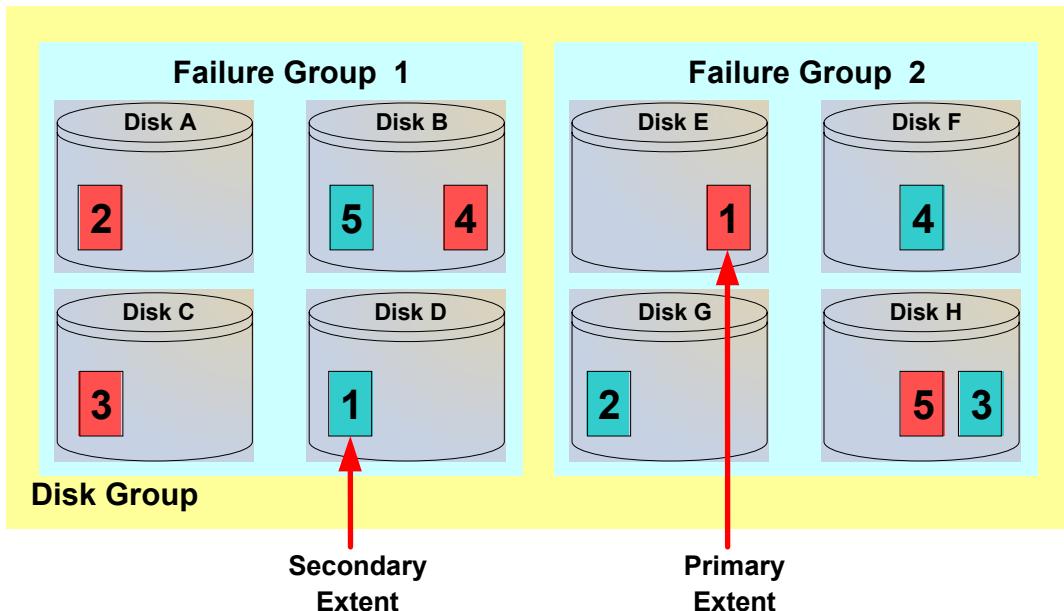
디스크 그룹 내에서 디스크는 Failure 그룹으로 수집될 수 있습니다. Failure 그룹은 저장 영역 또는 데이터베이스 관리자가 ASM Mirroring이 수행되는 하드웨어 경계를 지정하는 방법입니다. 예를 들어, 단일 디스크 컨트롤러에 연결된 모든 디스크는 공통 Failure 그룹 내에서 지정될 수 있습니다. 이렇게 하면 별개의 컨트롤러에 연결된 디스크에서 파일 Extent가 Mirroring됩니다. 또한 관리자는 ASM을 구성하여 기본 Failure 그룹 정책을 선택할 수 있습니다. 기본 정책은 개별 디스크가 고유한 Failure 그룹에 있도록 하는 것입니다.

필요한 모든 조건을 사용하여 Failure 그룹으로 디스크를 그룹화할 수 있습니다. Failure 그룹을 사용하면 개별 디스크, 디스크 컨트롤러, I/O 네트워크 구성 요소 및 심지어 전체 저장 영역 시스템까지 Failure로부터 보호할 수 있습니다. 일반적으로 관리자는 자신의 저장 영역 환경을 분석하고 특정 Failure 시나리오의 문제를 완화할 수 있도록 Failure 그룹을 구성합니다.

자신의 설치에서 가장 적합한 Failure 그룹 구성을 선택하는 것은 데이터베이스 또는 저장 영역 관리자에게 달려 있습니다.

## 스트라이핑 및 Mirroring 예제

총 8개의 디스크를 포함하고 2개의 Failure 그룹에 분산되어 있는 Normal redundancy 디스크 그룹



ORACLE

Copyright © 2010, Oracle. All rights reserved.

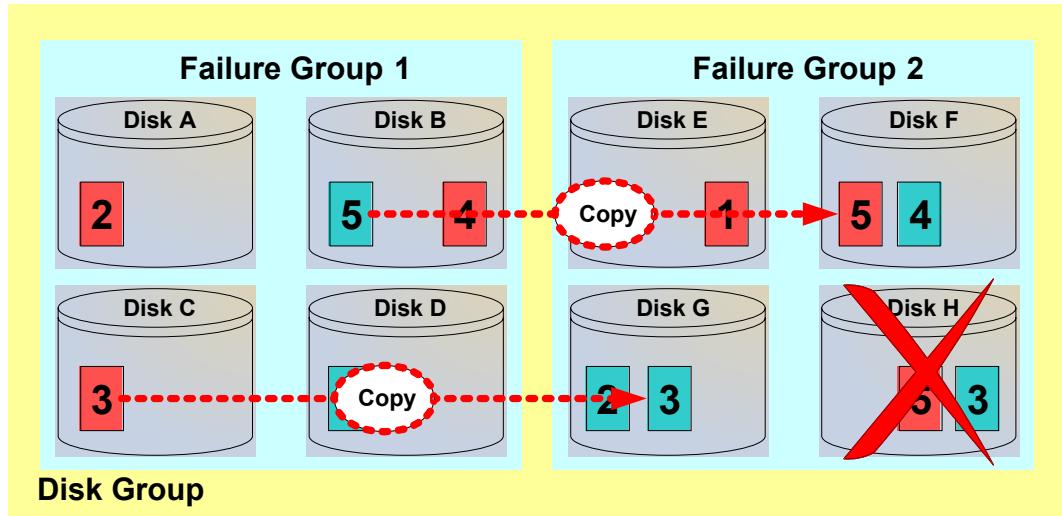
### 스트라이핑 및 Mirroring 예제

이 페이지의 다이어그램에서는 Normal redundancy 디스크 그룹의 스트라이핑 및 mirroring을 나타내는 예제를 보여줍니다. 빨간색 블록은 디스크 그룹의 8개 디스크 중 5개에 걸쳐 5개의 Extent가 스트라이핑되어 있는 파일을 나타냅니다. 파란색 블록은 파일 Extent의 mirroring된 복사본을 나타냅니다. 여러 디스크 및 Failure 그룹에 걸쳐 분산된 Extent에 관계없이 각 Extent는 각 Failure 그룹에서 정확히 한 개의 복사본을 갖습니다.

파일이 할당되면 성능을 위한 기본 Extent 및 무결성을 위한 보조 복사본이 할당됩니다. 따라서 모든 데이터베이스 읽기는 기본적으로 기본 Extent에서 수행됩니다.

## Failure 예제

디스크 H가 실패하면, 여기에 포함된 Extent가 정상적인 mirror에 있는 정상적인 디스크에서 재생성됩니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

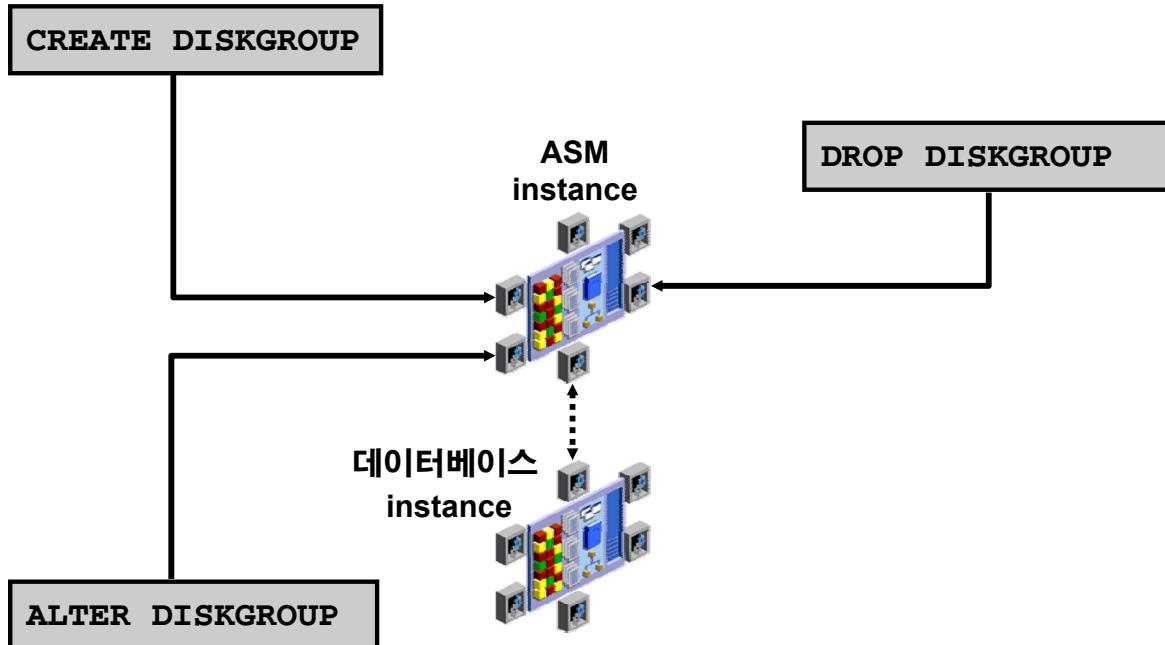
### Failure 예제

이전 페이지에서의 예제를 확장하여, 디스크 H가 실패하고 여기에 포함된 데이터에 더 이상 액세스할 수 없다고 가정해 보십시오. 이 Failure에서는 실패한 디스크의 모든 Extent를 Recovery하고 다른 디스크로 복사해야 합니다.

Extent 3과 5는 정상적인 복사본에서 동일 Failure 그룹에 있는 다른 디스크의 사용 가능한 영역에 복사됩니다. 이 예제에서 Extent 5는 디스크 B에서 디스크 F로 복사되고 Extent 3은 디스크 C에서 디스크 G로 복사됩니다.

디스크 실패 후 마지막 단계는 ASM이 디스크 그룹에서 실패한 디스크를 삭제하는 것입니다. 디스크 제거는 본질적으로 동일한 프로세스를 트리거하지만, 이 경우 제거하려는 디스크의 Extent가 먼저 사용 가능한 대체 위치로 복사됩니다.

# 디스크 그룹 관리



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 디스크 그룹 관리

ASM Instance의 주된 목적은 디스크 그룹을 관리하고 데이터를 보호하는 것입니다. 또한 ASM Instance는 파일 레이아웃을 데이터베이스 Instance에 전달합니다. 이런 방식으로 데이터베이스 Instance는 디스크 그룹에 저장된 파일에 직접 액세스할 수 있습니다.

디스크 그룹 관리 명령에는 여러 가지가 있습니다. 이러한 명령은 모두 SYSASM 또는 SYSDBA 권한이 필요하며 ASM Instance에서 실행되어야 합니다.

새 디스크 그룹을 추가할 수 있습니다. 또한 기존 디스크 그룹을 수정하여 새 디스크를 추가하고, 기존 디스크를 제거하며, 많은 다른 작업을 수행할 수도 있습니다. 기존 디스크 그룹을 제거할 수도 있습니다.

# SQL\*Plus를 사용하여 디스크 그룹 생성 및 삭제

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ sqlplus / AS SYSASM
SQL*Plus: Release 11.2.0.1.0 - Production on Wed Jul 8 20:46:46 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

..
SQL> CREATE DISKGROUP dgroupA NORMAL REDUNDANCY
FAILGROUP controller1 DISK
  '/devices/A1' NAME diskA1 SIZE 120G FORCE,
  '/devices/A2',
FAILGROUP controller2 DISK
  '/devices/B1',
  '/devices/B2';
```

```
SQL> DROP DISKGROUP dgroupA INCLUDING CONTENTS;
```

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 디스크 그룹 생성 및 삭제

ASM 디스크 검색으로 /dev 디렉토리에서 A1, A2, B1 및 B2 디스크가 식별되었다고 가정해 보십시오. 또한 A1 및 A2 디스크는 B1 및 B2 디스크의 개별 디스크 컨트롤러에 있다고 가정해 보십시오. 슬라이드의 첫번째 예제는 DGROUPA라는 디스크 그룹을 CONTROLLER1과 CONTROLLER2라는 두 개의 Failure 그룹으로 구성하는 방법을 보여줍니다.

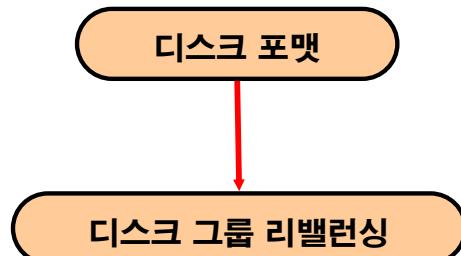
또한 예제에서는 디스크 그룹에 대해 기본 중복성 속성인 NORMAL REDUNDANCY를 사용합니다. 선택적으로 디스크 이름 및 크기를 지정할 수 있습니다. 이러한 정보를 지정하지 않으면 ASM은 기본 이름을 생성하고 디스크 크기를 결정하려고 합니다. 크기를 계산할 수 없으면 오류가 발생합니다. FORCE는 지정된 디스크가 이미 ASM 디스크 그룹의 멘버로 포맷되었더라도 지정된 디스크 그룹에 추가되어야 함을 나타냅니다. ASM 디스크 그룹의 멘버로 포맷되지 않은 디스크에 FORCE 옵션을 사용하면 오류가 반환됩니다.

슬라이드의 두번째 명령문에 표시된 대로 디스크 그룹을 모든 파일과 함께 삭제할 수 있습니다. 실수로 삭제하지 않도록 하려면 디스크 그룹에 내부 ASM 메타 데이터와 함께 모든 파일이 계속 포함되어 있는 경우 INCLUDING CONTENTS 옵션을 지정해야 합니다. 디스크 그룹은 삭제할 순서대로 마운트해야 합니다. 디스크 그룹 파일 중 아무것도 열리지 않도록 하면 그룹과 모든 드라이브가 디스크 그룹에서 제거됩니다. 그런 다음 각 디스크의 헤더를 겹쳐서 ASM 형식 정보를 제거합니다.

## 디스크 그룹에 디스크 추가

```
ALTER DISKGROUP dgroupA ADD DISK
  '/dev/sde1' NAME A5,
  '/dev/sdf1' NAME A6,
  '/dev/sdg1' NAME A7,
  '/dev/sdh1' NAME A8;
```

```
ALTER DISKGROUP dgroupA ADD DISK '/devices/A*' ;
```



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 디스크 그룹에 디스크 추가

이 예제에서는 디스크 그룹에 디스크를 추가하는 방법을 보여줍니다. 디스크를 추가하려면 ALTER DISKGROUP ADD DISK 명령을 실행합니다. 첫번째 명령문은 DGROUPA 디스크 그룹에 네 개의 새 디스크를 추가합니다.

두번째 명령문은 검색 문자열의 상호 작용을 보여줍니다. 다음 구성은 고려하십시오.

/devices/A1은 DGROUPA 디스크 그룹의 멤버입니다.

/devices/A2는 DGROUPA 디스크 그룹의 멤버입니다.

/devices/A3은 DGROUPA 디스크 그룹의 멤버입니다.

/devices/A4는 후보 디스크입니다.

두번째 명령은 A4를 DGROUPA 디스크 그룹에 추가합니다. 다른 디스크는 이미 DGROUPA 디스크 그룹의 일부이기 때문에 검색 문자열과 일치하는 경우에도 무시합니다. 다이어그램에 표시된 대로 디스크 그룹에 디스크를 추가하면 ASM Instance는 디스크의 주소를 지정하고 사용 가능하게 합니다. 그러면 디스크가 포맷되고 리밸런싱됩니다. 리밸런스 프로세스는 모든 파일의 Extent를 새 디스크로 이동하므로 시간이 많이 걸립니다.

**주:** 리밸런싱은 데이터베이스 작업을 차단하지 않습니다. 리밸런스 프로세스는 주로 시스템의 I/O 로드에 영향을 줍니다. 리밸런스 성능이 높을수록 시스템에 더 많은 I/O 로드가 적재됩니다. 따라서 낮은 I/O 대역폭은 데이터베이스 I/O에 사용할 수 있습니다.

## 기타 ALTER 명령

dgroupA에서 디스크 제거:

```
ALTER DISKGROUP dgroupA DROP DISK A5;
```

단일 명령에서 디스크 추가 및 삭제:

```
ALTER DISKGROUP dgroupA
  DROP DISK A6
  ADD FAILGROUP controller3
    DISK '/dev/sdi1' NAME A9;
```

디스크 삭제 작업 취소:

```
ALTER DISKGROUP dgroupA UNDROP DISKS;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 기타 ALTER 명령

슬라이드의 첫번째 명령문은 DGROUPA 디스크 그룹에서 디스크 중 하나를 제거하는 방법을 보여줍니다. 두번째 명령문은 단일 명령으로 디스크를 추가하고 삭제할 수 있는 방법을 보여줍니다. 이런 경우 가장 큰 이점은 명령이 완료될 때까지 리밸런싱이 시작되지 않는다는 점입니다. 세번째 명령문은 디스크 삭제 작업을 취소하는 방법을 보여줍니다. UNDROP 명령은 디스크 삭제가 보류된 경우에만 실행되며 완료된 삭제 작업에는 영향을 주지 않습니다.

다음 명령문은 필요한 경우 DGROUPB 디스크 그룹을 리밸런싱합니다.

```
ALTER DISKGROUP dgroupB REBALANCE POWER 5;
```

이 명령은 디스크가 추가, 삭제 또는 크기 조정될 때 자동으로 수행되기 때문에 대개 필요하지 않습니다. 그러나 POWER 절을 사용하여 ASM\_POWER\_LIMIT 초기화 파라미터로 정의한 기본 속도를 재정의하려는 경우에 유용합니다. 새 레벨로 명령을 다시 입력하면 지속적인 리밸런스 작업의 파워 레벨을 변경할 수 있습니다. 파워 레벨이 0이면 명령이 암시적으로나 명시적으로 다시 호출될 때까지 리밸런싱이 중지됩니다. 다음 명령문은 DGROUPA를 마운트 해제합니다.

```
ALTER DISKGROUP dgroupA DISMOUNT;
```

MOUNT 및 DISMOUNT 옵션을 통해 하나 이상의 디스크 그룹을 데이터베이스 Instance에서 사용 가능하거나 사용 불가능하도록 설정할 수 있습니다. 수동 마운트 해제 및 마운트 기능은 단일 Instance를 지원하는 클러스터화된 ASM 환경에서 해당 Instance가 다른 노드로 failover되는 경우에 유용합니다.

# Enterprise Manager를 사용한 ASM 관리

**General**

Current Status: Up  
Up Since: Jul 8, 2009 10:18:28 AM GMT+07:00  
Availability (%): 76.38 (Last 24 hours)

Instance Name: +ASM  
Version: 11.2.0.1.0  
Host: edrsr25p1.us.oracle.com  
Oracle Home: /u01/app/oracle/product/11.2.0/grid

**Disk Group Usage (GB)**

Disk Group	Size (GB)	Type
FRA	9.00	Unallocated
DATA	4.50	Internal

**Diagnostic Summary**

Alert Log: No ORA-errors  
Active Incidents: 0

**Serviced Databases**

Name	Disk Groups	Failure Groups	Allocated Space (GB)	Availability	Alerts
orcl.example.com	FRA, DATA	4 (0 down)	3.97	1	1 0
CLUSTER_UNKNOWN	DATA	4 (0 down)	0		Not Monitored

**Serviced ASM Cluster File Systems**

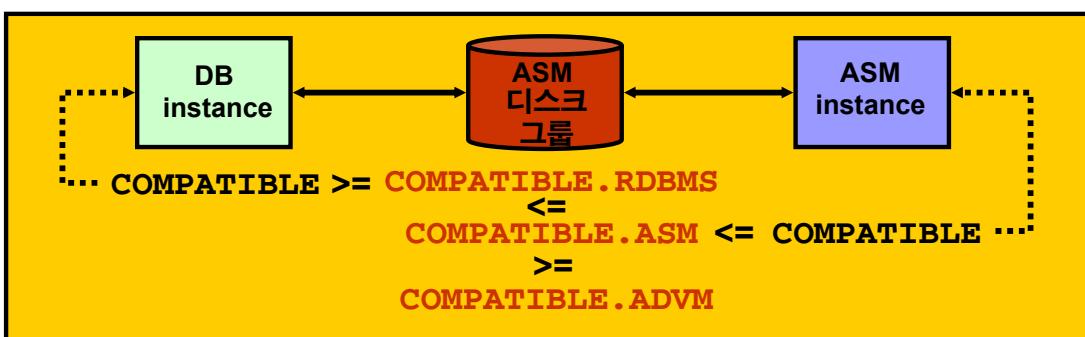
Mount Point	Availability	State	Used (%)	Used (GB)	Size (GB)	Allocated Space (GB)	Volume	Disk Group
(No ASM Cluster File Systems)								

## Enterprise Manager를 사용한 ASM 관리

Oracle EM(Enterprise Manager)은 데이터베이스, 미들웨어, 응용 프로그램, 네트워크, IT Infrastructure 등을 포함하는 오라클 관리 도구 제품군입니다. Enterprise Manager는 일반적인 ASM 관리 작업을 위한 포인트 앤 클릭 대체 도구를 제공하는 브라우저 기반 환경을 제공합니다.

# ASM 디스크 그룹 호환성

- 각 디스크 그룹의 호환성을 별도로 제어 가능:
  - ASM 호환성은 ASM 메타 데이터 온디스크(on-disk) 구조를 제어합니다.
  - RDBMS 호환성은 최소 소비자 클라이언트 레벨을 제어합니다.
  - ADVM 호환성은 디스크 그룹이 Oracle ASM 볼륨을 포함할 수 있는지 여부를 결정합니다.
- 디스크 그룹 호환성은 설정한 후에 취소할 수 없습니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASM 디스크 그룹 호환성

ASM 디스크 그룹에 적용할 수 있는 호환성 종류는 세 가지 즉, 디스크 그룹을 설명하는 persistent 데이터 구조 처리, 클라이언트의 호환성(디스크 그룹 소비자), 디스크 그룹의 볼륨을 포함하는 기능이 있습니다. 이러한 속성은 각각 *ASM 호환성*, *RDBMS 호환성* 및 *ADVM 호환성*이라고 부릅니다. 각 디스크 그룹의 호환성은 별도로 제어할 수 있습니다. 이는 Oracle Database 10g 및 Oracle Database 11g의 디스크 그룹이 혼합된 이기종 환경을 활성화하는 데 필수적입니다. 이 세 호환성 설정은 각 ASM 디스크 그룹의 속성입니다.

- RDBMS 호환성은 Instance가 디스크 그룹을 마운트할 수 있는 RDBMS Instance의 최소 호환 버전을 나타냅니다. 이 호환성은 ASM과 데이터베이스(RDBMS) Instance 간에 교환되는 메시지의 형식을 지정합니다. ASM Instance는 각기 다른 호환성 설정으로 실행되는 서로 다른 RDBMS 클라이언트를 지원할 수 있습니다. 각 Instance의 데이터베이스 호환 버전 설정은 해당 데이터베이스에서 사용되는 모든 디스크 그룹의 RDBMS 호환성보다 크거나 같아야 합니다. 데이터베이스 Instance는 일반적으로 ASM Instance가 아닌 Oracle 흘에서 실행됩니다. 즉, 데이터베이스 Instance가 ASM Instance와 다른 소프트웨어 버전을 실행할 수도 있습니다. 데이터베이스 Instance가 ASM Instance에 처음 연결할 때는 둘 다 지원하는 가장 높은 버전을 조정합니다.

## ASM 디스크 그룹 호환성(계속)

데이터베이스의 호환성 파라미터 설정, 데이터베이스의 소프트웨어 버전, 디스크 그룹의 RDBMS 호환성 설정에 따라 데이터베이스 Instance가 특정 디스크 그룹을 마운트할 수 있는지 여부가 결정됩니다.

- ASM 호환성은 디스크에서 ASM 메타 데이터의 데이터 구조 형식을 제어하기 위한 Persistent 호환성 설정을 나타냅니다. 디스크 그룹의 ASM 호환성 레벨은 항상 동일한 디스크 그룹의 RDBMS 호환성 레벨보다 크거나 같아야 합니다. ASM 호환성은 ASM 메타 데이터의 형식과만 관련이 있습니다. 파일 컨텐트의 형식은 데이터베이스 Instance에 의해 결정됩니다. 예를 들어, 디스크 그룹의 ASM 호환성은 11.0으로 설정되어 있고 해당 RDBMS 호환성은 10.1일 수 있습니다. 즉, 이 디스크 그룹은 소프트웨어 버전이 11.0 이상인 ASM 소프트웨어를 통해서만 관리될 수 있지만 소프트웨어 버전이 10.1 이상인 모든 데이터베이스 클라이언트가 해당 디스크 그룹을 사용할 수 있습니다.
- ADVM 호환성 속성은 디스크 그룹에 Oracle ASM 볼륨이 포함될 수 있는지 여부를 결정합니다. 값은 11.2 이상으로 설정되어야 합니다. 이 속성을 설정하기 전에 COMPATIBLE\_ASM 값은 11.2 이상이어야 합니다. 또한 ADVM 볼륨 드라이버가 로드되어 있어야 합니다.

디스크 그룹의 호환성은 Persistent 디스크 구조나 프로토콜 메시징에 변화가 있는 경우에만 높여야 합니다. 그러나 디스크 그룹 호환성 높이기는 취소할 수 없는 작업입니다. 디스크 그룹 호환성은 CREATE DISKGROUP 명령 또는 ALTER DISKGROUP 명령을 사용하여 설정할 수 있습니다.

**주:** 디스크 그룹 호환성 외에도 *compatible* 파라미터(데이터베이스 호환 버전)에 따라 활성화되는 기능이 결정됩니다. 이 파라미터는 *instance\_type* 파라미터에 따라 데이터베이스 또는 ASM Instance에 적용됩니다. 예를 들어, 이 파라미터를 10.1로 설정하면 Oracle Database 11g에 도입된 기능(디스크 온라인/오프라인, 가변 Extent 등)이 배제됩니다.

## ASM 디스크 그룹 속성

이름	속성	값	설명
au_size	Create, Alter	1 2 4 8 16 32 64MB	디스크 그룹의 할당 단위 (AU: Allocation Unit) 크기
compatible.rdbms	Create, Alter	유효한 데이터베이스 버전	DB와 ASM 사이에 교환되는 메시지의 형식
compatible.asm	Create, Alter	유효한 ASM Instance 버전	디스크에 있는 ASM 메타 데이터 구조의 형식
compatible.advm	Create, Alter	유효한 ASM Instance 버전	디스크 그룹에서 Oracle ASM 볼륨 허용
disk_repair_time	Create, Alter	0 M to 2 <sup>32</sup> D	OFFLINE으로 설정된 후 디스크를 제거할 때까지의 시간
template.tname.redundancy	Alter	UNPROTECT MIRROR HIGH	지정된 템플리트의 중복성
template.tname.stripes	Alter	COARSE FINE	지정된 템플리트의 스트라이핑 속성

```
CREATE DISKGROUP DATA2 NORMAL REDUNDANCY
DISK '/dev/sda1', '/dev/sdb1'
ATTRIBUTE 'compatible.asm'='11.2';
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### ASM 디스크 그룹 속성

ASM 디스크 그룹을 생성하거나 변경할 때마다 CREATE DISKGROUP 및 ALTER DISKGROUP 명령의 새 ATTRIBUTE 절을 사용하여 해당 속성을 변경할 수 있습니다. 이 속성은 슬라이드에 제공된 테이블에 간략하게 요약되어 있습니다.

- ASM은 디스크 그룹을 생성할 때 여러 할당 단위(AU: Allocation Unit) 크기를 지정할 수 있도록 합니다. AU의 크기는 1, 2, 4, 8, 16, 32 또는 64MB일 수 있습니다.
  - RDBMS 호환성: 자세한 내용은 "ASM 디스크 그룹 호환성" 섹션을 참조하십시오.
  - ASM 호환성: 자세한 내용은 "ASM 디스크 그룹 호환성" 섹션을 참조하십시오.
  - DISK\_REPAIR\_TIME은 분(M), 시간(H) 또는 일(D) 단위로 지정할 수 있습니다. 단위를 생략하면 기본적으로 H가 사용되고, 속성을 생략하면 기본값으로 3.6H가 사용됩니다.
- 이 속성은 ALTER DISKGROUP ... DISK OFFLINE 문을 사용하여 재정의할 수 있습니다.
- 지정된 템플리트의 중복성 속성을 지정할 수 있습니다.
  - 지정된 템플리트의 스트라이핑 속성을 지정할 수 있습니다.

주: 정의된 각 디스크 그룹에 대해 V\$ASM\_ATTRIBUTE 고정 뷰를 통해 정의된 모든 속성을 확인할 수 있습니다.

# Enterprise Manager를 사용하여 디스크 그룹 속성 편집

**Current Disk Group Usage (GB)**

**Edit Advanced Attributes for Disk Group: DATA**

**Disk Group Compatibility**

Advancing the disk group compatibility enables the user to use new features available in the newer version. This operation can not be reversed.

Database Compatibility	10.1.0.0.0
ASM Compatibility	11.2.0.0.0
ASM Volume Compatibility	
Disk Repair Time (Hours)	3.6
Smart Scan Capability	Disabled
File Access Control	Disabled

**TIP** The database compatibility has to be less than or equal to the ASM compatibility. The ASM Volume compatibility can only be set when ASM compatibility is 11.2 and above.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Enterprise Manager를 사용하여 디스크 그룹 속성 편집

Enterprise Manager를 사용하면 디스크 그룹과 관련된 환경 설정을 손쉽게 저장하고 검색할 수 있습니다.

Create Disk Group 페이지 및 Edit Disk Group Advanced Attributes 페이지에서 호환되는 속성을 설정할 수 있습니다. `disk_repair_time` 속성은 Edit Disk Group Advanced Attributes 페이지에만 추가됩니다.

**주:** 11g 이전 ASM Instance의 경우 기본 ASM 호환성 및 클라이언트 호환성은 모두 10.1입니다. 11g ASM Instance의 경우 기본 ASM 호환성은 11.2이고 데이터베이스 호환성은 10.1입니다.

## ASM 메타 데이터 검색

- SQL\*Plus 사용:

```
SQL> SELECT f.type, f.redundancy, f.striped, f.modification_date,
a.system_created, a.name FROM v$asm_alias a, v$asm_file f WHERE
a.file_number = f.file_number and a.group_number = f.group_number
and type='DATAFILE';

TYPE      REDUND  STRIPE  MODIFICAT   S  NAME
-----  -----  -----  -----  -  -----
DATAFILE  MIRROR  COARSE  08-JUL-09  Y  SYSTEM.256.689832921
DATAFILE  MIRROR  COARSE  08-JUL-09  Y  SYSAUX.257.689832923
..
```

- asmcmd 사용:

```
ASMCMD> ls -l +DATA/orcl/datafile
Type      Redund  Striped  Time                  Sys  Name
DATAFILE  MIRROR  COARSE   JUL 08 21:00:00  Y  SYSTEM.256.689832921
DATAFILE  MIRROR  COARSE   JUL 08 21:00:00  Y  SYSAUX.257.689832923
..
```

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### ASM 메타 데이터 검색

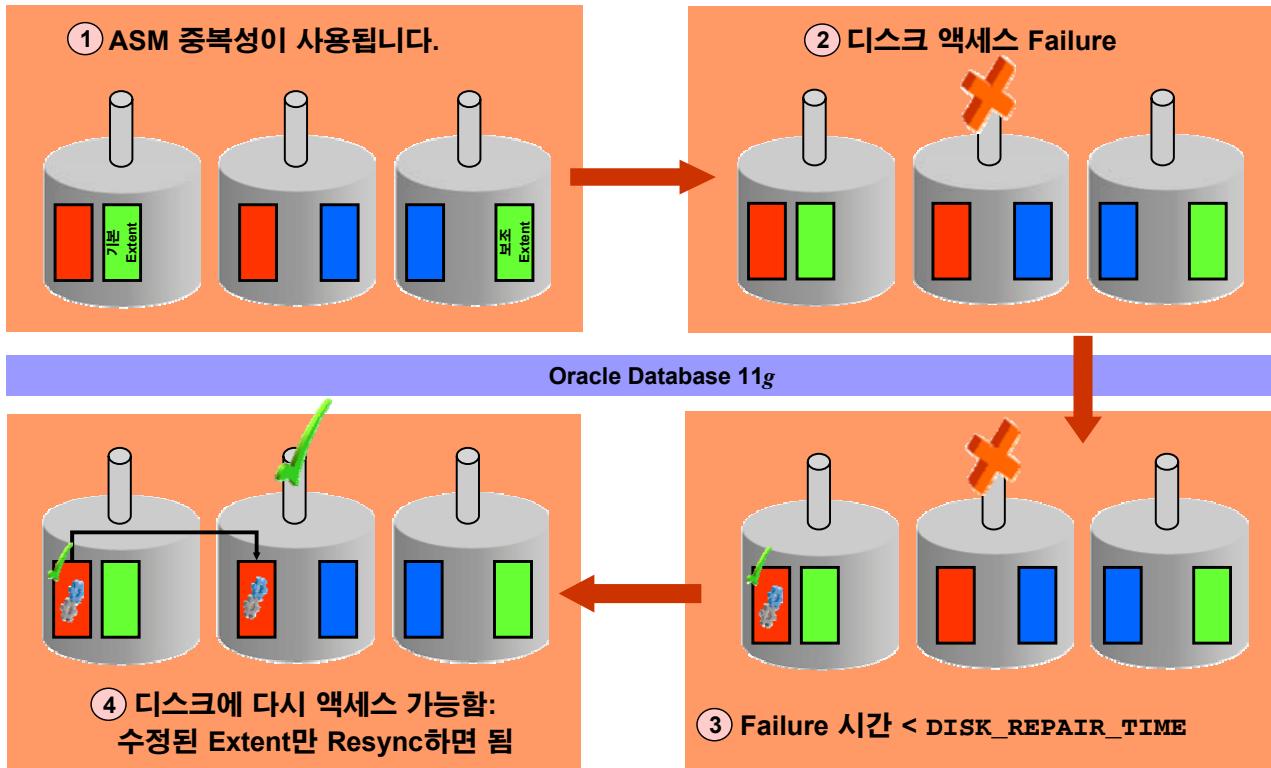
ASM Instance는 Dynamic Performance 뷰로 제공된 메모리 기반 메타 데이터 테이블을 호스트합니다. 이 데이터는 SQL\*Plus, asmcmd 또는 Enterprise Manager를 사용하여 query할 수 있습니다.

SQL\*Plus를 사용하려면 SQL 언어에 대한 지식이 필요하며, 관련 정보 검색을 위해 여러 Dynamic Performance 뷰를 조인해야 할 수 있습니다. 슬라이드에서 이 첫번째 예제에서는 데이터베이스의 데이터 파일과 관련된 메타 데이터 표시를 위한 v\$asm\_file 및 v\$asm\_alias 사이의 조인이 표시됩니다. 이 query를 ASM Instance에 대해 수행할 경우 구문이 설계된 방식에 따라 여러 데이터베이스에서 데이터 파일을 검색할 수 있습니다. 출력을 단일 데이터베이스로 제한하도록 추가 필터 조건을 사용할 수 있습니다.

asmcmd 유ти리티는 SQL 언어에 대한 지식 없이도 ASM Instance에 연결하여 메타 데이터를 검색할 수 있는 장점이 있습니다. 이 유ти리티는 UNIX 표기와 비슷한 스타일을 사용합니다. 이 슬라이드의 두번째 예제에서는 asmcmd를 사용하여 SQL 예제와 동일한 메타 데이터를 검색합니다. 이 예제의 또 다른 장점은 나열되는 경로에 orcl의 데이터베이스 이름 및 datafile의 파일 유형이 포함되기 때문에 단일 데이터베이스의 데이터 파일로 출력이 제한된다는 점입니다. 따라서 asmcmd에서 디렉토리로 표시되는 내용을 위해서는 동일 결과를 제공하도록 WHERE 절을 사용하는 SQL 필터 조건이 필요합니다.

주: Enterprise Manager Database Control은 단순히 여러 ASM 웹 페이지 사이를 이동하여 대부분의 ASM 메타 데이터를 표시할 수 있습니다.

## ASM Fast Mirror Resync 개요



Copyright © 2010, Oracle. All rights reserved.

ORACLE

### ASM Fast Mirror Resync 개요

ASM Fast Mirror Resync를 사용하면 디스크의 Transient Failure를 재동기화하는 데 필요한 시간이 크게 줄어듭니다. Transient Failure 후에 디스크가 오프라인 상태가 되면 ASM은 정전되어 있는 동안 수정된 Extent를 추적합니다. Transient Failure가 복구되면 ASM은 정전 시 영향을 받은 ASM 디스크 Extent만 신속하게 재동기화할 수 있습니다.

이 기능에서는 영향을 받은 ASM 디스크의 내용이 손상되거나 수정되지 않았다고 가정합니다. ASM 디스크 경로가 실패하면 ASM 디스크가 오프라인 상태가 되지만 해당 디스크 그룹에 대한 DISK\_REPAIR\_TIME 속성을 설정한 경우에는 삭제되지 않습니다. 이러한 속성 설정에 따라 복구 완료 후 재동기화할 수 있는 상태를 유지하면서 ASM이 허용하는 디스크 정전의 지속 시간이 결정됩니다.

**주:** 추적 메커니즘은 수정된 각 할당 단위(AU: Allocation Unit)에 대해 1비트를 사용하여 추적 메커니즘의 효율성을 극대화합니다.

## 퀴즈

ASM Instance에는 어떤 파라미터가 필요합니까?

1. **INSTANCE\_TYPE**
2. **ASM\_DISKGROUPS**
3. **LARGE\_POOL\_SIZE**
4. 위 항목 중 답 없음

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1

## 퀴즈

하위단계(Fine-grain) 스트라이핑은 기본적으로 \_\_\_\_\_  
및 \_\_\_\_\_에 사용됩니다.

- 1. 데이터 파일**
- 2. 콘트롤 파일**
- 3. 임시 파일**
- 4. 온라인 리두 로그**
- 5. SPFILE**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2, 4

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- SQL\*plus의 asmcmd 및 Enterprise Manager를 사용하여 ASM Instance 관리
- ASM 디스크 그룹 생성 및 삭제
- ASM 호환성 속성 지정
- ASM 디스크 그룹 확장
- ASM 메타 데이터 검색 방법 비교



Copyright © 2010, Oracle. All rights reserved.

## 연습 5 개요: ASM Instance 관리

이 연습에서는 다음 내용을 다룹니다.

- **asmcmd**
- **로 ASM 디스크 그룹 생성EM으로 ASM 디스크 그룹 삭제**
- **ASM 메타 데이터 보기**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# Oracle 네트워크 환경 구성

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- Enterprise Manager를 사용하여 다음 작업 수행
  - 추가 리스너 생성
  - Oracle Net 서비스 alias 생성
  - Connect-time failover 구성
  - Oracle Net 리스너 제어
- tnsping을 사용하여 Oracle Net 연결 테스트
- Shared Server를 사용하는 경우와 Dedicated Server를 사용하는 경우 식별

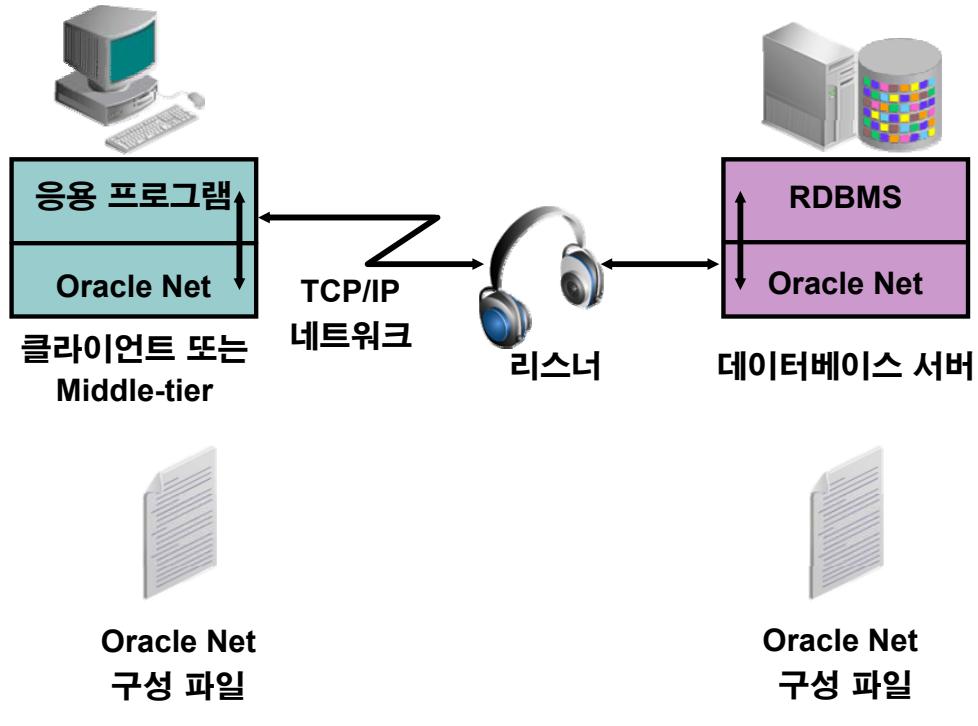
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 리소스

- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database Net Services Reference*

# Oracle Net 서비스



Copyright © 2010, Oracle. All rights reserved.

ORACLE

## Oracle Net 서비스

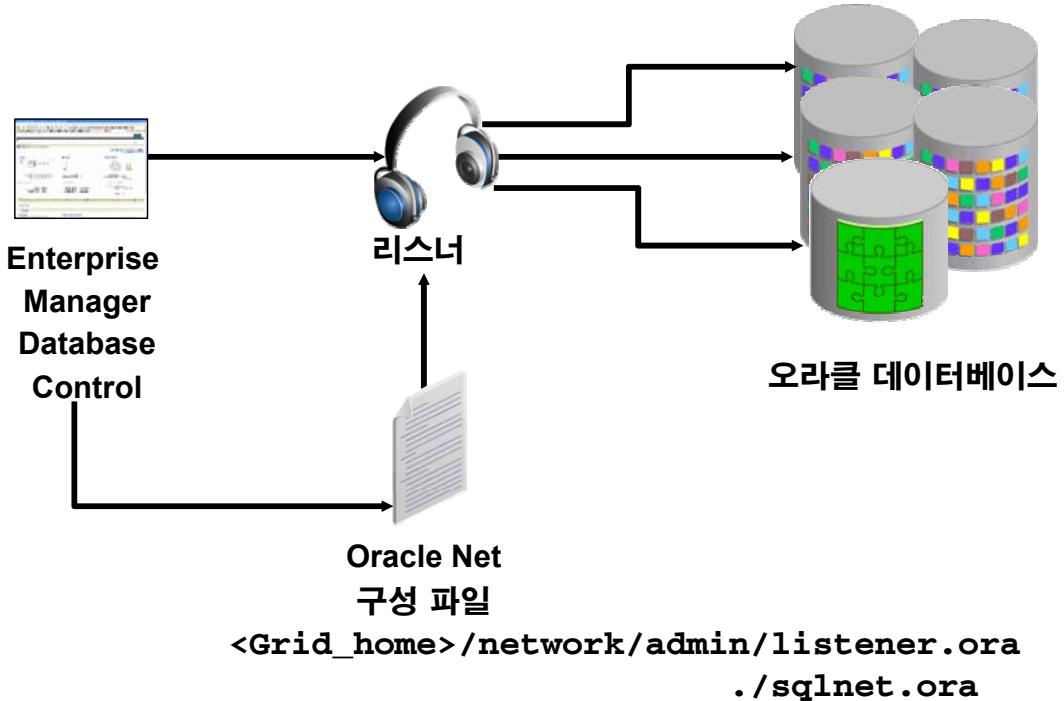
Oracle Net 서비스는 클라이언트 또는 Middle-tier 응용 프로그램에서 Oracle 서버로의 네트워크 연결을 활성화합니다. 네트워크 세션이 설정되면 Oracle Net은 클라이언트 응용 프로그램과 데이터베이스 서버 모두에 대해 데이터 전달 수단으로 사용됩니다. Oracle Net은 클라이언트 응용 프로그램 및 데이터베이스 서버 간의 메시지 교환뿐만 아니라 연결을 설정하고 유지 관리해야 합니다. Oracle Net 또는 Oracle Net를 시뮬레이트하는 JDBC(Java Database Connectivity) 등은 데이터베이스 서버와 통신해야 하는 각 컴퓨터에 위치하고 있습니다.

클라이언트 컴퓨터에서 Oracle Net은 데이터베이스에 대한 응용 프로그램 연결의 백그라운드 구성 요소입니다.

데이터베이스 서버의 Oracle Net은 데이터베이스와 외부 응용 프로그램 간의 연결을 조정하는 *Oracle Net 리스너*라는 활성 프로세스를 포함합니다.

Oracle Net 서비스는 일반적으로 들어오는 데이터베이스 연결을 허용하는 데 사용됩니다. 외부 코드 라이브러리(EXTPROC)에 대한 액세스를 허용하고 Oracle Heterogeneous Services를 통해 Oracle Instance를 비오라클 데이터 소스(예: Sybase, Informix, DB2 및 SQL Server)에 연결하도록 추가 네트 서비스를 구성할 수 있습니다.

# Oracle Net 리스너



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Oracle Net 리스너

Oracle Net 리스너(또는 리스너)는 로컬이 아닌 모든 유저 연결을 위한 Oracle Instance의 게이트웨이입니다. 단일 리스너는 다중 데이터베이스 instance 및 수천 개의 클라이언트 연결을 지원할 수 있습니다.

Enterprise Manager는 리스너에 액세스하는 방법 중 하나입니다. 암호 보호 및 로그 파일 위치와 같은 일반 파라미터는 물론 실제 리스너 구성도 제어할 수 있습니다.

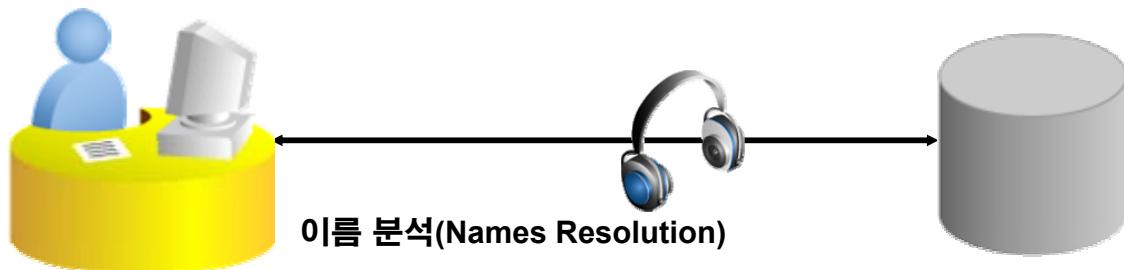
또한 고급 관리자는 필요한 경우 vi 또는 gedit와 같은 표준 운영 체제(OS) 텍스트 편집기로 구성 파일을 수동으로 편집하여 Oracle Net 서비스를 구성할 수 있습니다.

**주:** 독립 서버에 대한 그리드 Infrastructure 설치 시, Oracle Net 리스너는 자체 소프트웨어 설치 디렉토리인 `<Grid_home>`에서 시작됩니다. ASM Instance에 연결 기능을 제공하려면 리스너를 이 소프트웨어 설치에서 실행해야 합니다. 이 리스너는 또한 기본적으로 동일 서버에 설치된 모든 데이터베이스 instance를 수신하는 데 사용됩니다.

## Net 연결 설정

Oracle Net에서 클라이언트 또는 Middle-tier 연결을 설정하려면  
클라이언트는 다음 사항을 알아야 합니다.

- 리스너가 실행 중인 호스트
- 리스너가 모니터 중인 포트
- 리스너가 사용 중인 프로토콜
- 리스너가 처리 중인 서비스 이름



ORACLE

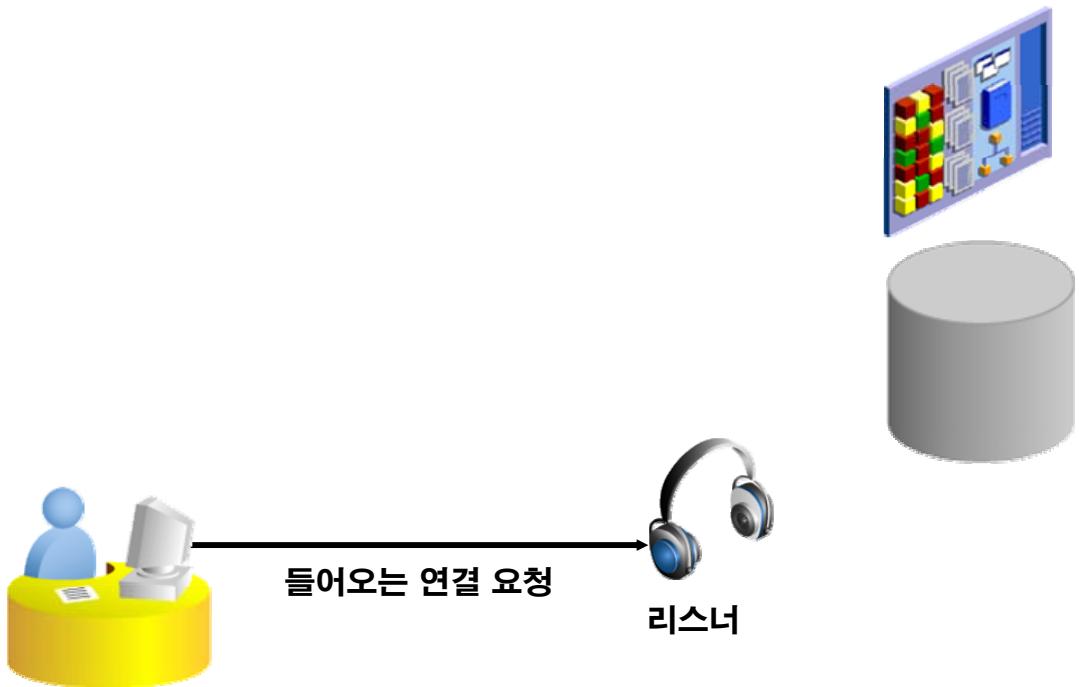
Copyright © 2010, Oracle. All rights reserved.

### Net 연결 설정

응용 프로그램이 Oracle Net 리스너를 통해 서비스에 연결하려면 리스너가 있는 주소 또는 호스트, 리스너가 적용할 프로토콜, 리스너가 모니터 중인 포트를 포함하여 해당 서비스에 대한 정보가 있어야 합니다. 리스너가 배치된 후 응용 프로그램에서 필요로 하는 정보의 마지막 부분은 응용 프로그램이 연결하려는 서비스 이름입니다.

*Oracle Net 이름 분석(Names Resolution)*은 이 연결 정보를 확인하는 절차입니다.

## 연결 설정



ORACLE®

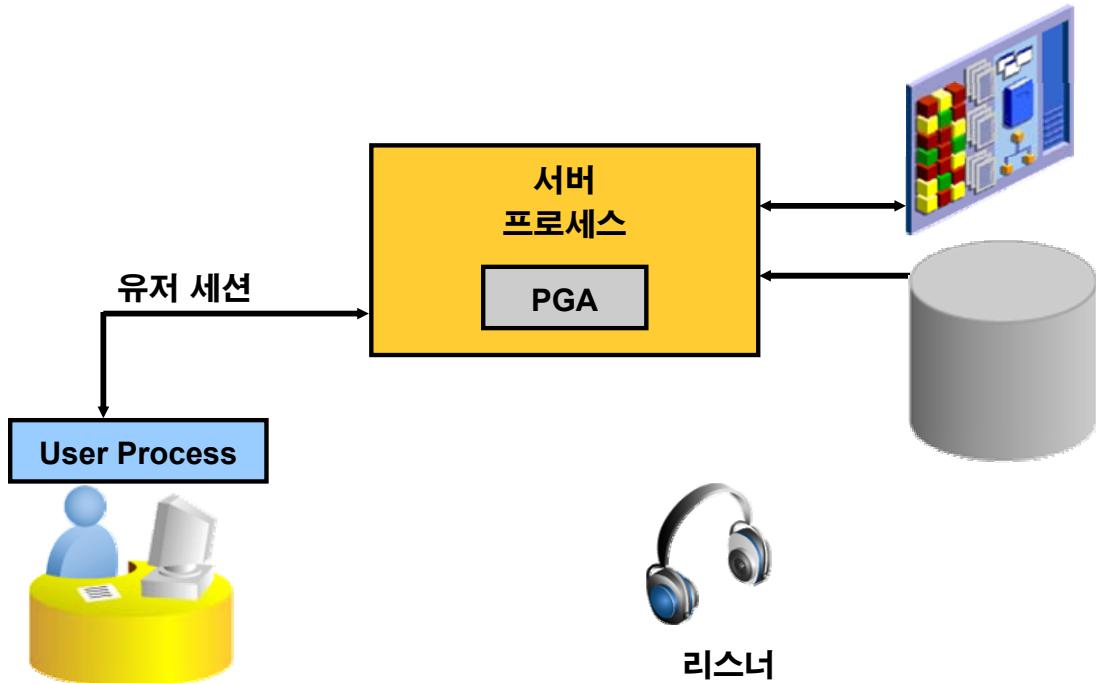
Copyright © 2010, Oracle. All rights reserved.

### 연결 설정

Oracle Net 이름 분석(Names Resolution)이 완료되면 유저 또는 Middle-tier 응용 프로그램(이후 *user process*라고 함)의 연결 요청이 리스너로 전달됩니다. 리스너는 CONNECT 패킷을 수신한 다음 이 CONNECT 패킷이 적합한 Oracle Net 서비스 이름을 요청하고 있는지 검사합니다.

`tnsping` 요청과 같이 서비스 이름을 요청하지 않는 경우 리스너는 연결 요청을 확인하기만 합니다. 부적합한 서비스 이름을 요청한 경우 리스너는 User Process에 오류 코드를 전송합니다.

# 유저 세션



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 유저 세션

CONNECT 패킷이 적합한 서비스 이름을 요청한 경우 리스너는 연결을 처리할 새 프로세스를 생성합니다. 이 새 프로세스를 **서버 프로세스**라고 합니다. 리스너는 이 프로세스에 연결하여 User Process의 주소 정보를 포함하여 초기화 정보를 전달합니다. 이 시점부터 리스너는 더 이상 연결을 처리하지 않고 모든 작업을 서버 프로세스로 전달합니다.

서버 프로세스는 유저 인증서(일반적으로 암호)를 확인하며 이 인증서가 적합한 경우 유저 세션이 생성됩니다.

**Dedicated server 프로세스:** 세션이 설정되면 서버 프로세스가 이제 서버에서 유저 에이전트로 작동합니다. 서버 프로세스는 다음 작업을 수행합니다.

- 응용 프로그램을 통해 실행된 모든 SQL 문을 구문 분석하고 실행합니다.
- SQL 문을 수행하는 데 필요한 데이터 블록의 데이터베이스 버퍼 캐시를 검사합니다.
- SGA(System Global Area)에 데이터 블록이 없는 경우 디스크의 데이터 파일에서 SGA의 데이터베이스 버퍼 캐시로 필요한 데이터 블록을 읽어 들입니다.
- 모든 정렬 작업을 관리합니다. 정렬 영역은 정렬 작업에 사용되는 메모리 영역으로, 메모리에서 PGA(Program Global Area)와 관련된 부분에 들어 있습니다.
- 응용 프로그램이 정보를 처리할 수 있는 방식으로 User Process에 결과를 반환합니다.
- 감사(audit) 옵션을 읽고 감사 대상에 User Process를 보고합니다.

# Oracle 네트워크 구성 및 관리 도구

- **Enterprise Manager Net Services Administration 페이지**
- **Oracle Net Manager**
- **Oracle Net Configuration Assistant**
- **명령행**



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Oracle 네트워크 구성 및 관리 도구

Oracle 네트워크 구성을 관리하려면 다음 도구 및 응용 프로그램을 사용하십시오.

- **Enterprise Manager:** 통합된 환경에서 Oracle Net 서비스를 구성하고 관리할 수 있습니다. Enterprise Manager를 사용하여 다중 파일 시스템에서 Oracle 홈에 대한 Oracle Net 서비스를 구성하고 리스너를 관리할 수 있습니다.
- **Oracle Net Manager:** 로컬 클라이언트나 서버 호스트에서 Oracle 홈에 대한 Oracle Net 서비스를 구성할 수 있는 GUI(그래픽 유저 인터페이스)를 제공합니다. Oracle Net Manager를 사용하면 로컬 클라이언트 또는 서버 호스트에서 Oracle 홈에 대한 Oracle Net 서비스를 구성할 수 있습니다. Oracle Net Manager를 사용하면 다음 네트워크 구성 요소를 구성할 수 있습니다.
  - **이름 지정:** 간단한 이름과 연결 식별자를 정의하고 이를 Connect Descriptor로 매핑하여 서비스의 네트워크 위치와 ID를 식별합니다. Oracle Net Manager는 로컬 tnsnames.ora 파일 또는 중앙화된 디렉토리 서비스에서 Connect Descriptor의 구성을 지원합니다.
  - **이름 지정 방식:** 연결 식별자가 Connect Descriptor로 분석되는 다양한 방법을 구성합니다.
  - **프로파일:** 클라이언트 또는 서버에서 Oracle Net 기능을 활성화 및 구성하기 위한 환경 설정을 구성합니다.
  - **리스너:** 클라이언트 연결을 수신할 리스너를 생성하고 구성합니다.

## Oracle 네트워크 구성 및 관리 도구(계속)

- **Oracle Net Configuration Assistant:** Oracle 소프트웨어를 설치할 때 Oracle Universal Installer에 의해 실행됩니다. Oracle Net Configuration Assistant를 사용하면 오라클 데이터베이스에 대한 수신 프로토콜 주소와 서비스 정보를 구성할 수 있습니다. 일반적인 데이터베이스 설치를 수행하는 동안 Oracle Net Configuration Assistant는 데이터베이스에 대한 TCP/IP 수신 프로토콜 주소가 지정된 LISTENER라는 리스너를 자동으로 구성합니다. 커스텀 설치를 수행하는 경우 Oracle Net Configuration Assistant에서는 선택한 리스너 이름과 프로토콜 주소를 구성하라는 메시지를 표시합니다. 데이터베이스를 설치한 후 초기 네트워크 구성은 위해 Oracle Net Configuration Assistant를 사용하십시오. 그런 다음 Oracle Enterprise Manager 및 Oracle Net Manager를 사용하여 네트워크를 구성 및 관리할 수 있습니다.
- **명령행:** 리스너 프로세스의 상태를 시작 및 정지하고 보는 데 사용됩니다. OS 유저 (이 과정에서는 oracle)가 리스너를 시작하거나 정지합니다. 리스너가 시작되지 않은 경우 Enterprise Manager를 사용할 수 없습니다.

# Listener Control 유틸리티

**Oracle Net 리스너는 lsnrctl 명령행 유틸리티를 사용하거나 EM에서 제어할 수 있습니다.**

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
$ lsnrctl
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 30-JUN-2009 00:47:01
Copyright (c) 1991, 2009, Oracle. All rights reserved.
Welcome to LSNRCTL, type "help" for information.

LSNRCTL> help
The following operations are available
An asterisk (*) denotes a modifier or extended command:
start          stop          status
services       version       reload
save_config    trace         spawn
change_password quit         exit
set*           show*

```

Copyright © 2010, Oracle. All rights reserved.

## Listener Control 유틸리티

Instance가 시작되면 리스너 프로세스가 오라클 데이터베이스에 대한 통신 경로를 설정합니다. 그런 다음 리스너가 데이터베이스 연결 요청을 받아들일 수 있습니다.

Listener Control 유틸리티를 사용하여 리스너를 제어할 수 있습니다. lsnrctl을 사용하여 다음 작업을 수행할 수 있습니다.

- 리스너 시작
- 리스너 정지
- 리스너 상태 검사
- 구성 파일 파라미터에서 리스너 다시 초기화
- 여러 리스너 동적 구성
- 리스너 암호 변경

이 유틸리티의 기본 명령 구문은 다음과 같습니다.

LSNRCTL> command [*listener\_name*]

lsnrctl 명령을 실행하면 다른 리스너 이름을 지정하거나 SET CURRENT\_LISTENER 명령을 실행하지 않는 한 기본 리스너(LISTENER)에서 명령이 실행됩니다. 리스너 이름이 LISTENER인 경우 *listener\_name* 인수는 생략할 수 있습니다. 슬라이드에서 lsnrctl에 유효한 명령이 나와 있습니다.

주: lsnrctl 유틸리티는 그리드 Infrastructure 홈 및 오라클 데이터베이스 홈에 있습니다.

lsnrctl 유틸리티를 사용하기 전 환경 변수를 적합한 홈으로 설정해야 합니다.

# Listener Control 유틸리티 구문

**Listener Control 유틸리티 명령은 명령행이나 LSNRCTL 프롬프트에서 실행할 수 있습니다.**

- **명령행 구문:**

```
$ lsnrctl <command name>
$ lsnrctl start
$ lsnrctl status
```

- **프롬프트 구문:**

```
LSNRCTL> <command name>
LSNRCTL> start
LSNRCTL> status
```

Copyright © 2010, Oracle. All rights reserved.

## Listener Control 유틸리티 구문

`lsnrctl` 명령은 유틸리티(프롬프트 구문) 또는 명령행 내에서 실행할 수 있습니다.

다음 두 명령은 효과가 같지만 각각 명령행 구문과 프롬프트 구문을 사용합니다.

### 명령행 구문:

```
$ lsnrctl start
```

### 프롬프트 구문:

```
$ lsnrctl
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on
30-JUN-2009 01:00:01
Copyright (c) 1991, 2009, Oracle. All rights reserved.
Welcome to LSNRCTL, type "help" for information.
LSNRCTL> start
```

일반적으로 명령행 구문은 개별 명령 또는 스크립트로 작성된 명령을 실행하는 데 사용됩니다.

`lsnrctl` 명령을 연속하여 여러 번 실행하려는 경우에는 프롬프트 구문이 더 효율적입니다.

`listener_name` 인수가 생략되었으므로 정지 명령이 LISTENER라는 리스너에 영향을 미칩니다. 리스너가 암호로 보호된 경우 프롬프트 구문을 사용해야 합니다.

## Listener Control 유틸리티 구문(계속)

리스너의 이름을 LISTENER 이외의 다른 이름으로 지정한 경우 명령에 리스너 이름을 포함하거나 SET CURRENT\_LISTENER 명령을 사용해야 합니다. 리스너의 이름을 custom\_lis로 지정했다고 가정합니다. 다음은 프롬프트 구문을 사용하여 custom\_lis라는 리스너를 정지하는 두 가지 예제입니다.

```
LSNRCTL> stop custom_lis
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host01)(PORT=5521)))
The command completed successfully
```

이 결과는 아래의 결과와 동일합니다.

```
LSNRCTL> set cur custom_lis
Current Listener is custom_lis
LSNRCTL> stop
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host01)(PORT=5521)))
The command completed successfully
```

**주:** 위 구문에서, current\_listener는 cur라는 약어로 표기되었습니다.

명령행 구문을 사용해도 동일한 결과가 나옵니다.

```
$ lsnrctl stop custom_lis
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on
30-JUN-2009 01:01:53
Copyright (c) 1991, 2009, Oracle. All rights reserved.
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host01)(PORT=5521)))
The command completed successfully
```

## SRVCTL을 사용하여 리스너 시작 및 정지

Oracle Restart가 리스너를 모니터하도록 구성된 경우,  
해당 리스너를 관리하려면 SRVCTL을 사용하십시오.

- 예제 구문:

```
$ srvctl -h  
$ srvctl start listener  
$ srvctl stop listener  
$ srvctl start listener -l mylistener  
$ srvctl status listener
```

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### SRVCTL을 사용하여 리스너 시작 및 정지

SRVCTL을 사용하여 Oracle Restart로 관리되는 모든 리스너를 시작 및 정지하고 상태를 볼 수 있습니다. 이에 대한 예는 다음과 같습니다.

- SRVCTL에서 제공되는 명령에 대한 도움말 표시: `srvctl -h`
- 기본 리스너 시작: `srvctl start listener`
- 기본 리스너 정지: `srvctl stop listener`
- `mylistener`라는 이름의 리스너 시작: `srvctl start listener -l mylistener`
- 기본 리스너 상태 표시: `srvctl status listener`

# Listener Home 페이지

The screenshot shows two tabs: 'General' and 'State'. A red arrow points from the 'Listener' entry in the General tab to the 'Alias' entry in the State tab.

General		State		
Status	Up	TNS Ping (ms)	10	
Up Since	Jun 19, 2009 2:02:00 AM GMT+07:00	Established Connections per minute	2.2	
Instance Name	orcl	Refused Connections per minute	0	
Version	11.2.0.1.0			
Host	<a href="#">edrsr25p1.us.oracle.com</a>			
Listener	<a href="#">LISTENER</a>			
ASM	<a href="#">+ASM_edrsr25p1.us.oracle.com</a>			
<a href="#">View All Properties</a>				
General		State		
Status	Up	TNS Ping (ms)	10	
Availability (%)	100	Established Connections per minute	2.2	
(Last 24 Hours)		Refused Connections per minute	0	
Alias	LISTENER			
Version	11.2.0.1.0			
Oracle Home	<a href="#">/u01/app/oracle/product/11.2.0/grid</a>			
Net Address	(ADDRESS=(PROTOCOL=TCP) (HOST=edrsr25p1.us.oracle.com)(PORT=1521))			
LISTENER.ORA Location	<a href="#">/u01/app/oracle/product/11.2.0/grid/network</a>			
	/admin			
Start Time	Jun 18, 2009 3:20:31 AM			
Host	<a href="#">edrsr25p1.us.oracle.com</a>			
Oracle Restart	Enabled			

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Listener Home 페이지

Enterprise Manager Database Home 페이지에서 리스너 링크를 누르면 Listener Home 페이지에 액세스할 수 있습니다.

이 페이지에서 다음을 확인할 수 있습니다.

- 최근 24시간 동안의 리스너 상태 및 가용성
- 리스너 버전 및 Oracle 흘
- 리스너의 첫번째 수신 주소
- 리스너를 시작하는 데 사용되는 구성 파일의 위치
- 리스너 시작 시간 및 호스트 정보
- Oracle Restart 상태

리스너를 시작하려면 Database Home 페이지에서 리스너 이름을 눌러 Listener Home 페이지를 엽니다. 실행 중인 리스너를 정지하려면 Stop을 누르고 실행되지 않은 리스너를 시작하려면 Start를 누릅니다. 리스너를 시작 및 정지할 수 있는 OS 유저로 호스트에 로그온합니다.

# Net Services Administration 페이지

The screenshot shows the Oracle Enterprise Manager 11g interface. At the top, it says "ORACLE Enterprise Manager 11g Database Control". Below that, it shows the host "Host edrsr25p1.us.oracle.com >". The main title is "Net Services Administration". It says "Net Services Administration allows you to configure or administer the following network components:" followed by a bulleted list. After the list, it says "Choose a configuration file location, then select the feature that you want to administer and click 'Go'." A dropdown menu is open, showing "Administer" and "Listeners". The "Listeners" option is highlighted with a red box. To the right of the dropdown, there's a table with columns "Feature", "Location", and "Oracle Home". The first row shows "Listener" with "11.2.0/db\_home1/network/admin" and "/u01/app/oracle/product/11.2.0/db\_home1". Below the table are links for "Database", "Setup", "Preferences", "Help", and "Logout".

## Net Services Administration 페이지

Net Services Administration 페이지에서 여러 파일 시스템의 Oracle 홈에 대해 Oracle Net 서비스를 구성할 수 있습니다. 이 페이지는 리스너에 대한 일반적인 관리 기능도 제공합니다. Net Services Administration을 사용하여 다음 항목을 구성 및 관리할 수 있습니다.

- 리스너:** 리스너를 추가, 제거, 시작 및 정지하는 것은 물론 추적 및 로깅 특성을 변경합니다. 또한 리스너의 제어 상태 보고서를 볼 수 있습니다.
- 디렉토리 이름 지정:** 간단한 이름과 연결 식별자를 정의하고 이를 Connect Descriptor로 매핑하여 서비스의 네트워크 위치와 ID를 식별합니다. 데이터베이스 서비스, Net 서비스 및 Net 서비스 alias를 중앙 디렉토리 서비스에 저장합니다.
- 로컬 이름 지정:** Net 서비스 이름을 tnsnames.ora 파일에 저장합니다.
- 프로파일:** sqlnet.ora 파라미터를 구성합니다.
- 파일 위치:** Net 서비스의 구성 파일 위치를 변경합니다.

## 리스너 생성

The screenshot shows the Oracle Net Services Administration interface with four numbered steps:

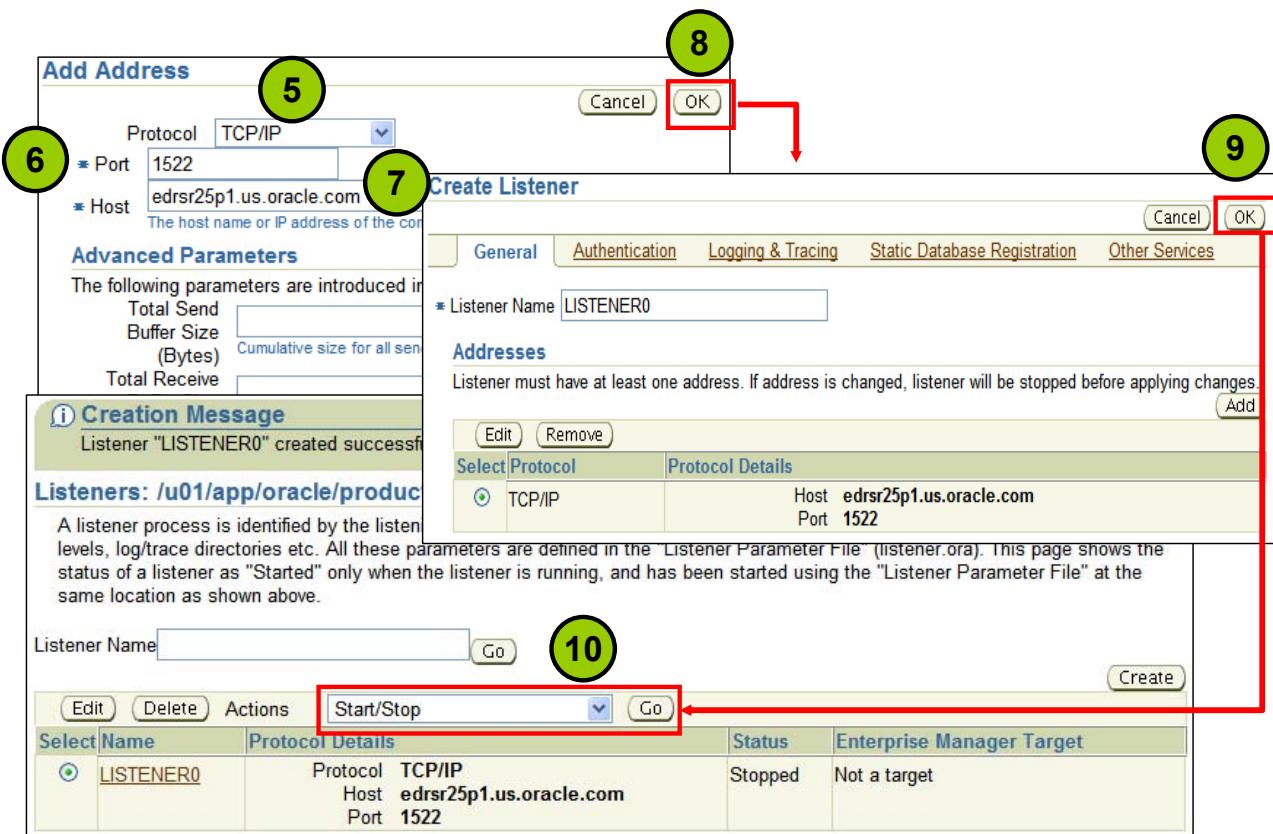
- 1.** In the "Net Services Administration" page, the "Administer Listeners" dropdown is selected. A red box highlights this dropdown.
- 2.** The "Host Login" dialog box is displayed. It shows the host as "edrsr25p1.us.oracle.com", Oracle Home as "/u01/app/oracle/product/11.2.0/db\_home1", and the user as "oracle". A red box highlights the "Go" button.
- 3.** The "Create Listener" dialog box is displayed. The "Listener Name" field contains "LISTENER0". A red box highlights the "Create" button.
- 4.** The "Listeners: /u01/app/oracle/product/11.2.0/db\_home1/network/admin" status page is shown. It displays information about the listener process, including host, port, and log/trace details. A red box highlights the "Add" button under the "Protocol Details" section.

### 리스너 생성

Oracle Net 리스너를 생성하려면 리스너 속성 페이지의 Related Links 영역에서 Net Services Administration을 누릅니다. 그런 후 다음 단계를 수행하십시오.

1. Administer drop-down list에서 Listeners를 선택하고 Go를 누릅니다.
2. 호스트에 대한 선호 인증서를 입력하지 않은 경우, Host Login 페이지가 나타납니다. username 및 암호를 입력하고 Login을 누릅니다.
3. Create를 누릅니다.
4. 리스너 이름을 입력합니다. 이름은 이 서버에서 고유해야 합니다. Add를 눌러 리스너 주소를 추가합니다. 각 리스너에는 적어도 하나의 리스너 주소가 있어야 합니다.

## 리스너 주소 추가



ORACLE

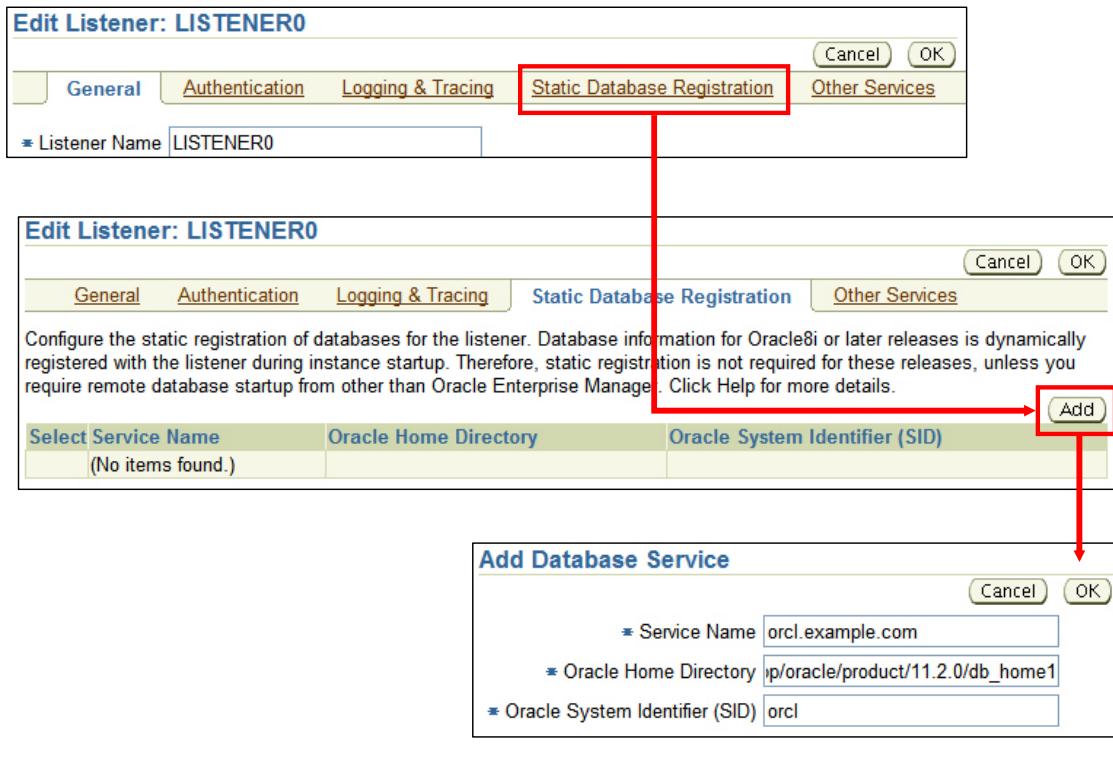
Copyright © 2010, Oracle. All rights reserved.

### 리스너 주소 추가

리스너 생성 워크플로우에 따라 리스너 주소 생성을 계속합니다.

5. 네트워크 프로토콜을 선택합니다. TCP/IP가 가장 많이 사용되고 있으며 기본값입니다. 일반적으로 데이터베이스 서버에 있는 로컬 응용 프로그램에 연결하는 데 사용되는 IPC(Internal Process Communication) 또는 외부 코드 라이브러리(EXTPROC) 및 SSL을 사용하는 TCP/IP를 선택할 수 있습니다.
- 주: EXTPROC 프로토콜은 Other Services 탭을 사용하여 구성합니다.
6. 리스너가 모니터할 포트를 입력합니다. Oracle Net의 기본 포트는 1521입니다. 1521 이외의 다른 포트를 사용하는 경우 리스너 또는 Instance에 대한 추가 구성이 필요합니다.
7. 리스너가 실행될 서버의 이름 또는 IP 주소를 입력합니다.
8. 리스너에 대한 다른 모든 구성 단계는 선택 사항입니다. OK를 눌러 주소를 저장합니다. 수신 주소와 이름만 구성하면 됩니다.
9. Create Listener 페이지에서, 방금 생성한 주소에 대한 정보를 검토하고 OK를 눌러 변경 사항을 저장합니다.
10. 새 리스너를 시작하려면 Actions drop-down list에서 Start/Stop을 선택한 다음 Go를 누릅니다.

# 데이터베이스 서비스 등록



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 서비스 등록

리스너가 instance로 클라이언트 연결을 전달하려면 리스너는 instance의 이름과 instance의 ORACLE\_HOME의 위치를 알아야 합니다. 리스너는 다음 두 가지 방법으로 이 정보를 찾을 수 있습니다.

- 동적 서비스 등록:** Oracle8i 이상 버전의 Instance는 데이터베이스 시작 시 자동으로 기본 리스너에 등록됩니다. 기본 리스너에 추가로 리스너를 구성할 필요가 없습니다.
- 정적 서비스 등록:** 이전 릴리스의 오라클 데이터베이스는 리스너에 자동으로 등록되지 않으므로 리스너가 지원하는 모든 데이터베이스 서비스의 리스트가 리스너 구성 파일에 포함되어 있어야 합니다. 그러나 다음의 경우에는 새 릴리스에서도 정적 서비스 등록을 선택할 수 있습니다.
  - 리스너의 위치가 기본 포트인 1521이 아니며 instance가 기본 포트에만 등록되도록 구성하려는 경우
  - 응용 프로그램이 정적 서비스 등록을 필요로 하는 경우

정적 데이터베이스 서비스를 추가하려면 Edit Listener 페이지에서 Static Database Registration을 누른 다음 Add 버튼을 누릅니다. 서비스 이름(global database name <DB\_NAME>. <DB\_DOMAIN>과 같은 방식), ORACLE\_HOME 경로 및 SID(instance 이름과 같은 방식)를 입력합니다. OK를 누릅니다. 변경 사항을 적용하려면 RELOAD 명령을 사용하여 리스너를 재로드하거나, 재시작해야 합니다.

## 데이터베이스 서비스 등록(계속)

### 서비스 이름

The SERVICE\_NAMES 초기화 파라미터는 클라이언트가 Instance에 연결할 수 있는 하나 이상의 이름을 지정합니다. Instance는 해당 서비스 이름을 리스너에 등록합니다. 클라이언트가 서비스를 요청하면 리스너는 요청된 서비스를 제공하는 Instance를 확인하고 해당 Instance에 대한 클라이언트의 경로를 지정합니다.

이 예제와 같이 각기 다르게 사용되는 동일한 데이터베이스를 구분하기 위해 여러 서비스 이름을 지정할 수 있습니다.

```
SERVICE_NAMES = sales.example.com, eurosales.example.com
```

또한 서비스 이름을 사용하여 복제(Replication)를 통해 서로 다른 두 데이터베이스에서 사용할 수 있는 단일 서비스를 식별할 수도 있습니다.

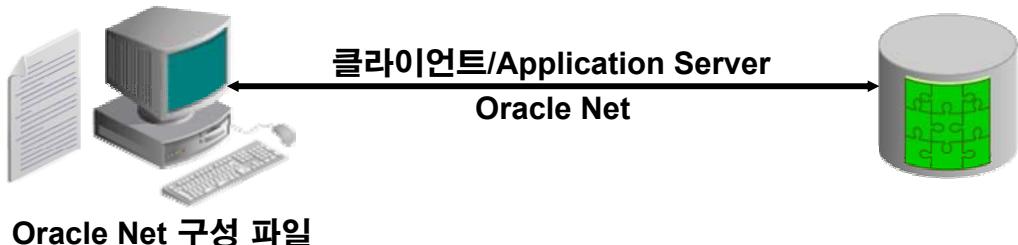
이 파라미터에서 이름을 도메인으로 한정하지 않으면 오라클에서 이를 DB\_DOMAIN 파라미터 값으로 한정합니다. DB\_DOMAIN이 지정되어 있지 않으면 한정되지 않은 SERVICE\_NAMES 값에 도메인이 적용되지 않습니다.

클라이언트 연결 요청을 처리하는 동안 리스너는 이 파라미터 값과 클라이언트 Connect Descriptor의 SERVICE\_NAME 파라미터 값을 일치시키려고 합니다. 클라이언트 Connect Descriptor가 SID 파라미터를 사용하는 경우 리스너는 값의 매핑을 시도하지 않습니다. 이 파라미터 값은 대개 초기화 파라미터 파일의 DB\_NAME 및 DB\_DOMAIN 파라미터를 조합(DB\_NAME.DB\_DOMAIN)하여 얻지만, 이 값에 클라이언트가 서비스를 식별하는 데 사용되는 유효한 이름이 포함될 수도 있습니다.

# 이름 지정 방식

Oracle Net는 여러 가지 연결 정보 분석 방법을 지원합니다.

- 간단한 연결(Easy Connect) 이름 지정: TCP/IP 연결 문자열 사용
- 로컬 이름 지정: 로컬 구성 파일 사용
- 디렉토리 이름 지정: 중앙화된 LDAP 호환 디렉토리 서버 사용
- 외부 이름 지정: 지원되는 비오라클 이름 지정 서비스 사용



Oracle Net 구성 파일

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 이름 지정 방식

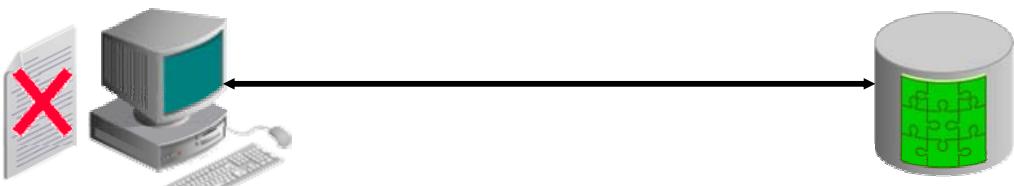
Oracle Net는 다음과 같은 이름 지정 방식을 지원합니다.

- 간단한 연결(Easy connect) 이름 지정: 이 방식을 사용하면 클라이언트는 다음과 같이 호스트 이름과 선택 사항인 포트 및 서비스 이름으로 구성된 TCP/IP 연결 문자열을 사용하여 오라클 데이터베이스 서버에 연결할 수 있습니다.  
CONNECT username/password@host[:port] [/service\_name]  
간단한 연결(Easy connect) 이름 지정 방식의 경우에는 구성이 필요 없습니다.
- 로컬 이름 지정: 로컬 이름 지정 방식은 Net 서비스 이름으로 식별되는 Connect Descriptor를 클라이언트의 로컬 구성 파일인 tnsnames.ora에 저장합니다.
- 디렉토리 이름 지정: 데이터베이스 서비스에 액세스하기 위해 디렉토리 이름 지정 방식에서는 연결 식별자를 LDAP(Lightweight Directory Access Protocol)와 호환되는 중앙화된 디렉토리 서버에 저장합니다.
- 외부 이름 지정: 이 방식은 네트 서비스 이름을 지원하는 비오라클 이름 지정 서비스에 저장합니다. 지원되는 Third-party 서비스는 다음과 같습니다.
  - NIS(Network Information Service) 외부 이름 지정
  - DCE(Distributed Computing Environment)의 CDS(Cell Directory Service)

## 간단한 연결(Easy Connect)

- 기본적으로 활성화됨
- 클라이언트측 구성이 필요 없음
- TCP/IP만 지원(SSL은 지원하지 않음)
- 다음과 같은 고급 연결 옵션은 지원하지 않음
  - Connect-time failover
  - 소스 경로 지정
  - 로드 밸런싱

```
SQL> CONNECT hr/hr@db.us.oracle.com:1521/dba11g
```



**Oracle Net 구성 파일 없음**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 간단한 연결(Easy Connect)

간단한 연결(Easy Connect)을 사용하여 연결 문자열의 일부로서 Oracle Net 연결에 필요한 모든 정보를 제공할 수 있습니다. 간단한 연결(Easy Connect)에서의 연결 문자열 형식은 다음과 같습니다.

<username>/<password>@<hostname>:<listener port>/<service name>

리스너 포트 및 서비스 이름은 선택 사항입니다. 리스너 포트를 입력하지 않으면 Oracle Net는 기본 포트인 1521을 사용하는 것으로 가정합니다. 서비스 이름을 입력하지 않으면 Oracle Net는 연결 문자열에 입력한 데이터베이스 서비스 이름과 호스트 이름이 동일한 것으로 가정합니다.

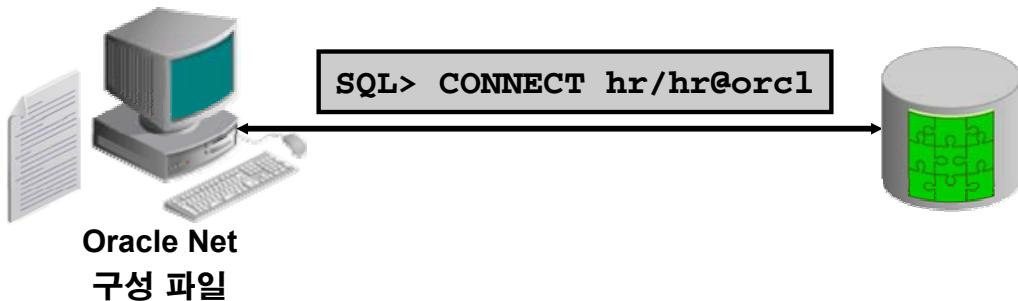
리스너가 포트 1521에서 수신하는 TCP와 SERVICE\_NAMES=db 및 DB\_DOMAIN=us.oracle.com Instance 파라미터를 사용한다고 가정하면 슬라이드에 나오는 연결 문자열을 다음과 같이 줄일 수 있습니다.

SQL> connect hr/hr@db.us.oracle.com

**주:** SERVICE\_NAMES 초기화 파라미터에 쉼표로 구분된 다중 값을 사용할 수 있습니다.  
이 시나리오가 제대로 작동하려면 이 값 중 하나만 db여야 합니다.

## 로컬 이름 지정

- 클라이언트측 이름 분석(Names Resolution) 파일 필요
- 모든 Oracle Net 프로토콜 지원
- 다음과 같은 고급 연결 옵션 지원
  - Connect-time failover
  - 소스 경로 지정
  - 로드 밸런싱



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 로컬 이름 지정

유저는 로컬 이름 지정을 사용하여 Oracle Net 서비스의 Alias를 제공합니다. Oracle Net는 alias를 알려진 서비스의 로컬 리스트와 비교하여 일치하는 것을 발견하면 alias를 호스트, 프로토콜, 포트 및 서비스 이름으로 변환합니다.

로컬 이름 지정의 장점 중 하나는 데이터베이스 유저가 간단한 연결(Easy Connect)에서 필요한 긴 연결 문자열 대신 짧은 alias만 기억하면 된다는 것입니다.

알려진 서비스의 로컬 리스트는 다음과 같은 텍스트 구성 파일에 저장됩니다.

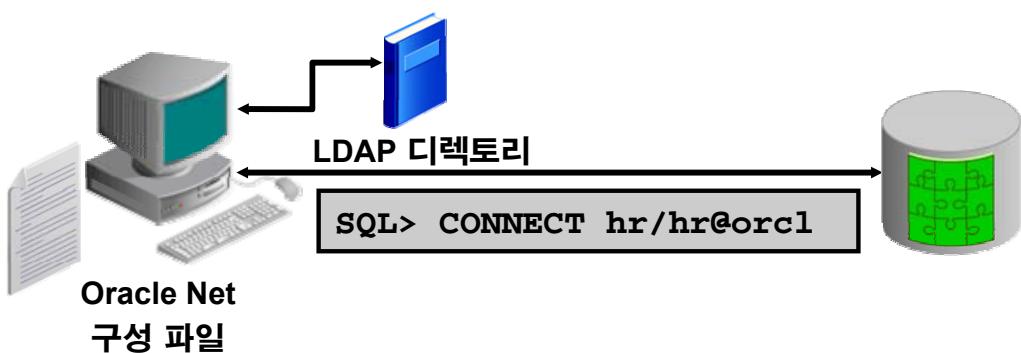
<oracle\_home>/network/admin/tnsnames.ora

이것은 tnsnames.ora 파일의 기본 위치지만 파일은 TNS\_ADMIN 환경 변수를 사용하여 어느 곳에든 배치할 수 있습니다.

로컬 이름 지정은 Oracle Net 서비스 구성이 자주 변경되지 않는 조직에 적합합니다.

## 디렉토리 이름 지정

- **로드된 Oracle Net 이름 분석(Names Resolution) 정보와 함께 LDAP 필요**
  - Oracle Internet Directory
  - Microsoft Active Directory Services
- **모든 Oracle Net 프로토콜 지원**
- **고급 연결 옵션 지원**



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 디렉토리 이름 지정

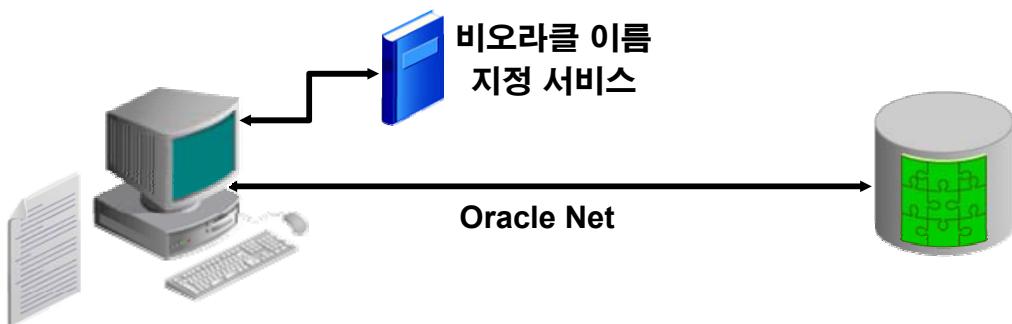
유저는 디렉토리 이름 지정을 사용하여 Oracle Net 서비스의 Alias를 제공합니다. Oracle Net는 alias를 알려진 서비스의 외부 리스트와 비교하여 일치하는 이름을 발견하면 alias를 호스트, 프로토콜, 포트 및 서비스 이름으로 변환합니다. 로컬 이름 지정처럼 데이터베이스 유저는 간단한 alias만 기억하면 됩니다.

디렉토리 이름 지정의 한 가지 장점은 새 서비스 이름이 LDAP 디렉토리에 추가되는 즉시 유저가 이 서비스 이름을 사용하여 연결할 수 있다는 것입니다. 로컬 이름 지정을 사용하는 경우 유저가 새 서비스 또는 수정된 서비스에 연결할 수 있으려면 데이터베이스 관리자(DBA)가 변경된 서비스 이름 정보가 포함되어 있는 개신된 tnsnames.ora 파일을 배포해야 합니다.

디렉토리 이름 지정은 Oracle Net 서비스 구성이 자주 변경되는 구성에 적합합니다.

## 외부 이름 지정 방식

- 지원되는 비오라클 이름 지정 서비스 사용
- 여기에는 다음이 포함됩니다.
  - NIS(Network Information Service) 외부 이름 지정
  - DCE(Distributed Computing Environment)의 CDS (Cell Directory Service)



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 외부 이름 지정 방식

외부 이름 지정 방식은 Net 서비스 이름을 지원되는 비오라클 이름 지정 서비스에 저장합니다.  
지원되는 Third-party 서비스는 다음과 같습니다.

- NIS(Network Information Service) 외부 이름 지정
- DCE(Distributed Computing Environment)의 CDS(Cell Directory Service)

외부 이름 지정은 디렉토리 이름 지정과 개념적으로 유사합니다.

# 서비스 Alias 구성

The screenshot shows the Oracle Net Services Administration interface. At the top, it says "Local Naming: /u01/app/oracle/product/11.2.0/db\_home1/network/admin". Below that, it says "These are the local Net Service Names in tnsnames.ora file at /u01/app/oracle/product/11.2.0/db\_home1/network/admin. You can test, edit, create and delete a Net Service Name." A "Search" field and a "Go" button are present.

A modal window titled "Create Net Service Name" is open. It has tabs for "General" and "Advanced". Under "General", the "Net Service Name" field contains "testorcl". The "Database Information" section explains that either a service name or SID must be provided. The "Use Service Name" option is selected, with "Service Name" set to "orcl.example.com". The "OK" button in this window is highlighted with a red box.

Another modal window titled "Add Address" is also open. It has tabs for "Protocol" and "Port". The "Protocol" dropdown is set to "TCP/IP", "Port" is set to "1522", and "Host" is set to "edrsr25p1.us.oracle.com". The "OK" button in this window is highlighted with a red box.

At the bottom left, there's a "Addresses" section with a "Select Protocol" dropdown and a "Protocol Details" table. An "Add" button is highlighted with a red box.

At the bottom right, the Oracle logo is visible.

## 서비스 Alias 구성

로컬 Oracle Net 서비스 Alias를 생성하려면 Net Services Administration 페이지의 Administer drop-down list에서 Local Naming을 선택하고 Go를 누릅니다. 그런 다음 Create를 누릅니다.

Local Naming 대신 Directory Naming을 선택하면 디렉토리 이름 지정에 대해 서비스 alias를 구성할 수 있습니다.

**주:** 디렉토리 이름 지정이 아직 구성되어 있지 않으면 Directory Naming 옵션을 선택할 수 없습니다. 디렉토리 이름 지정은 *Oracle Advanced Security Administration* 설명서 및 *Oracle Enterprise Identity Management* 과정에서 설명합니다.

Create Net Service Name 페이지의 Net Service Name 필드에 고유한 이름을 입력합니다. 이 이름은 유저가 이 Alias를 사용할 때 입력하는 이름입니다. 연결할 데이터베이스의 시스템 식별자(SID) 또는 서비스 이름을 입력하고 Add 버튼을 눌러 서비스 이름의 주소를 입력합니다.

주소에는 연결할 서비스의 리스너가 사용하는 프로토콜, 포트 및 호스트를 입력합니다.

## 고급 연결 옵션

**Oracle Net는 로컬 및 디렉토리 이름 지정과 함께 다음 고급 연결 옵션을 지원합니다.**

- **Connect-time failover**
- **소스 경로 지정**
- **로드 밸런싱**

Select	Protocol	Protocol Details
<input checked="" type="radio"/>	TCP/IP	Host edrsr25p1.us.oracle.com Port 1522
<input type="radio"/>	TCP/IP	Host edrsr25p1.us.oracle.com Port 1521

**Connect-time Failover and Client Load Balancing**

Configure whether addresses are tried randomly or sequentially during connections to the service. This setting is applicable only if there are more than one addresses configured.

- Try each address, in order, until one succeeds
- Try each address randomly, until one succeeds
- Try one address, selected at random
- Use each address in order until destination is reached
- Use only the first address

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 고급 연결 옵션

데이터베이스 서비스를 다중 리스너 프로토콜 주소를 통해 액세스할 수 있는 경우 주소를 사용할 순서를 지정할 수 있습니다. 주소는 무작위로 선택하거나 순차적으로 시도할 수 있습니다. Oracle RAC(Real Application Clusters) 구성과 같이 사용할 수 있는 리스너가 두 개 이상인 경우 Oracle Net는 리스너 Failover 및 로드 밸런싱은 물론 Oracle Connection Manager 소스 경로 지정을 활용할 수 있습니다.

*Connect-time failover*를 활성화하면 Alias는 두 개 이상의 나열된 리스너 주소를 갖습니다. 첫번째 주소에 연결할 수 없으면 두번째 주소에 연결을 시도합니다. Oracle Net는 제대로 작동하는 리스너에 도달하거나 모든 주소에 대한 연결이 실패할 때까지 나열된 순서대로 주소에 연결을 시도합니다. TAF(Transparent Application Failover)는 데이터베이스 Instance Failure 발생 시 클라이언트가 정상적인 데이터베이스에 재연결할 수 있도록 하는 클라이언트측 기능입니다. 서버가 클라이언트측에서 TAF 콜백을 트리거하는 데 통지가 사용됩니다.

로드 밸런싱을 활성화하면 Oracle Net는 주소 리스트에서 무작위로 주소를 선택합니다. 런타임 연결 로드 밸런싱 기능은 다중 디스패처 간에 활성 연결 수의 균형을 유지함으로써 연결 성능을 향상시킵니다. RAC 환경에서는 연결 풀 로드 밸런싱을 사용해도 다중 Instance 간에 활성 연결 수의 균형을 유지할 수 있습니다.

## 고급 연결 옵션(계속)

Oracle Connection Manager는 소스 경로 지정과 함께 Oracle Net 통신량에 대한 Proxy 서버로 사용되어 Oracle Net 통신량의 경로가 Firewall을 통해 안전하게 지정되도록 합니다. Oracle Net는 주소를 릴레이 리스트로 사용하여 대상에 도달할 때까지 첫번째 주소에 연결한 다음 두번째 주소로 전달해 줄 것을 요청합니다. 이것은 연결이 설정될 때마다 모든 주소가 사용된다는 점에서 failover 또는 로드 밸런싱과 다릅니다.

옵션	고급 기능
성공할 때까지 순서대로 주소를 하나씩 사용합니다.	Failover
성공할 때까지 주소를 무작위로 하나씩 사용합니다.	Failover 로드 밸런싱
무작위로 선택된 단일 주소를 사용합니다.	로드 밸런싱
대상에 도달할 때까지 순서대로 각 주소를 사용합니다.	소스 경로 지정
첫번째 주소만 사용합니다.	해당 사항 없음

## Oracle Net 연결 테스트

Oracle Net 서비스 alias를 테스트하는 tnsping 유틸리티:

- 클라이언트와 Oracle Net 리스너 간 연결 확인
- 요청된 서비스가 사용 가능한지 확인하지 않음
- 간단한 연결 이름 분석(Easy Connect Names Resolution) 지원

```
tnsping host01.example.com:1521/orcl
```

- 로컬 및 디렉토리 이름 지정 지원

```
tnsping orcl
```

ORACLE

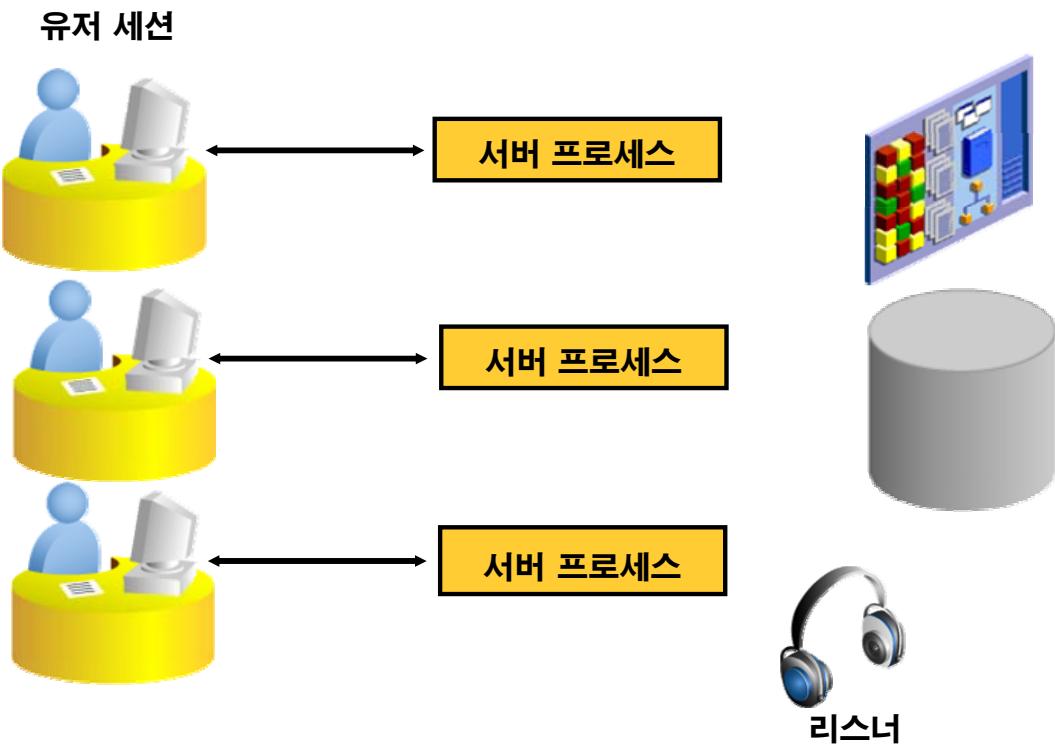
Copyright © 2010, Oracle. All rights reserved.

### Oracle Net 연결 테스트

tnsping은 TCP/IP의 펑(ping) 유틸리티와 동일한 Oracle Net 유틸리티입니다. 이 유틸리티를 사용하면 대상에 대한 네트워크 경로가 유효한지 신속하게 확인할 수 있습니다. 예를 들어, 명령행 window에서 tnsping orcl을 입력합니다.

이 유틸리티는 호스트 이름, 포트 및 프로토콜이 리스너에 도달하는지 확인합니다. 리스너가 서비스 이름을 처리하는지 실제로 검사하는 것은 아닙니다. 또한 tnsping 유틸리티를 사용하면 구성 파일의 위치를 알 수 있습니다. 다중 ORACLE\_HOME 위치를 가진 시스템에서 구성 파일의 위치를 알면 매우 유용합니다.

## 유저 세션: Dedicated Server 프로세스



Copyright © 2010, Oracle. All rights reserved.

**ORACLE**

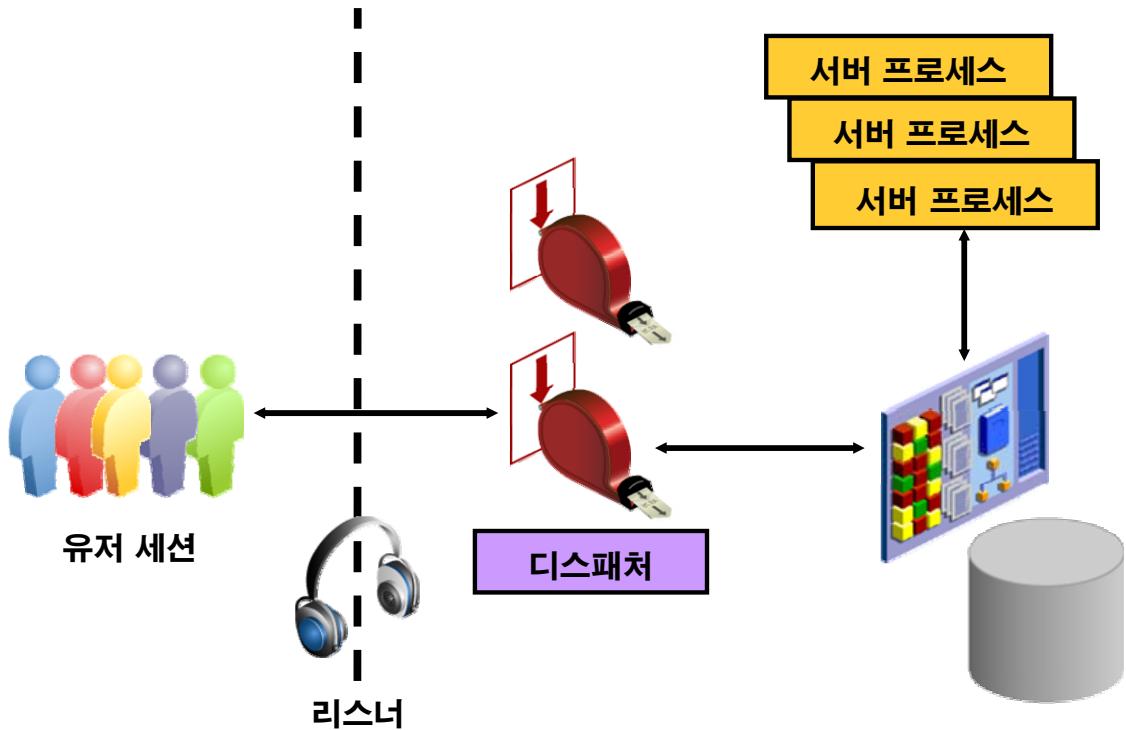
### 유저 세션: Dedicated Server 프로세스

Dedicated Server 프로세스에서는 한 개의 서버 프로세스에 한 개의 User Process가 연결됩니다. 각 서버 프로세스는 CPU 주기 및 메모리를 포함하는 시스템 리소스를 사용합니다.

시스템에 로드가 많아지면 Dedicated Server 프로세스가 사용하는 메모리 및 CPU 리소스가 많아지고 시스템 확장성에 악영향을 끼칠 수 있습니다. Dedicated Server 구조의 리소스 필요량이 많아 시스템에 악영향을 끼치는 경우 다음 두 가지 옵션이 있습니다.

- 메모리 및 CPU 용량을 늘려 시스템 리소스를 증가시킵니다.
- Oracle Shared Server 프로세스 구조를 사용합니다.

## 유저 세션: Shared Server 프로세스



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 유저 세션: Shared Server 프로세스

Shared Server 프로세스 구조와 관련된 각 서비스에는 디스패처 프로세스가 한 개(일반적으로 두 개 이상임) 이상 있습니다. 연결 요청이 도착하면 리스너가 Dedicated Server 프로세스를 생성하지 않고 각 디스패처에 대한 연결 로드(동시 연결 수)와 함께 각 서비스 이름에 사용할 수 있는 디스패처 리스트를 유지 관리합니다.

제공된 서비스 이름을 사용하는 디스패처 중 로드가 가장 적은 디스패처로 연결 요청 경로가 지정됩니다. 세션이 지속되는 동안 유저는 동일한 디스패처에 연결됩니다.

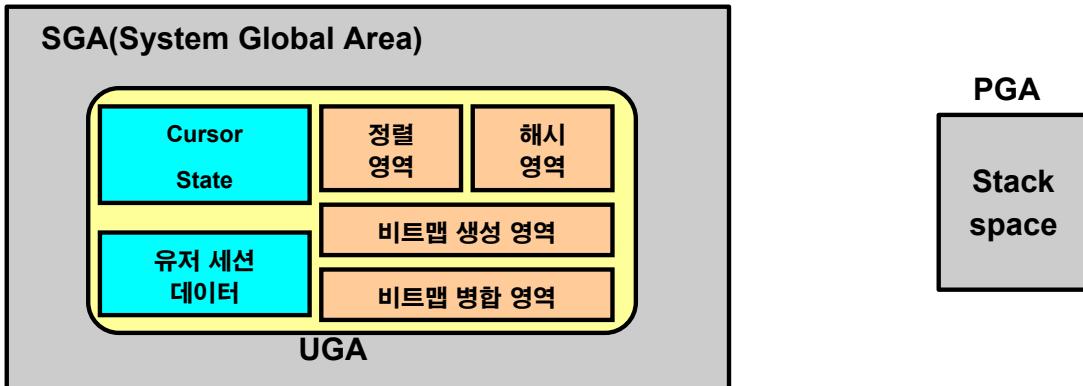
Dedicated Server 프로세스와 달리 단일 디스패처는 수백 개의 유저 세션을 관리할 수 있습니다. 디스패처는 유저 요청 작업을 실제로 처리하지는 않고 SGA의 Shared Pool 부분에 위치한 Common Queue에 유저 요청을 전달합니다.

Shared Server 프로세스는 이 Common Queue에서 요청을 가져온 다음 완료될 때까지 처리하여 Dedicated Server 프로세스 작업의 대부분을 넘겨 받습니다.

단일 유저 세션에 다중 Shared Server 프로세스에서 처리한 요청이 있을 수 있으므로 일반적으로 PGA에 저장된 대부분의 메모리 구조는 공유 메모리 위치(기본적으로 Shared Pool)에 두어야 합니다. 그러나 자동 메모리 관리에 대해 Large Pool이 구성되거나 SGA\_TARGET이 설정된 경우 이 메모리 구조는 SGA의 Large Pool 영역에 저장됩니다.

# SGA 및 PGA

Oracle Shared Server: 유저 세션 데이터가 SGA에 저장됩니다.



SGA의 크기를 조정하는 경우 Shared Server 메모리 요구 사항을 고려합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## SGA 및 PGA

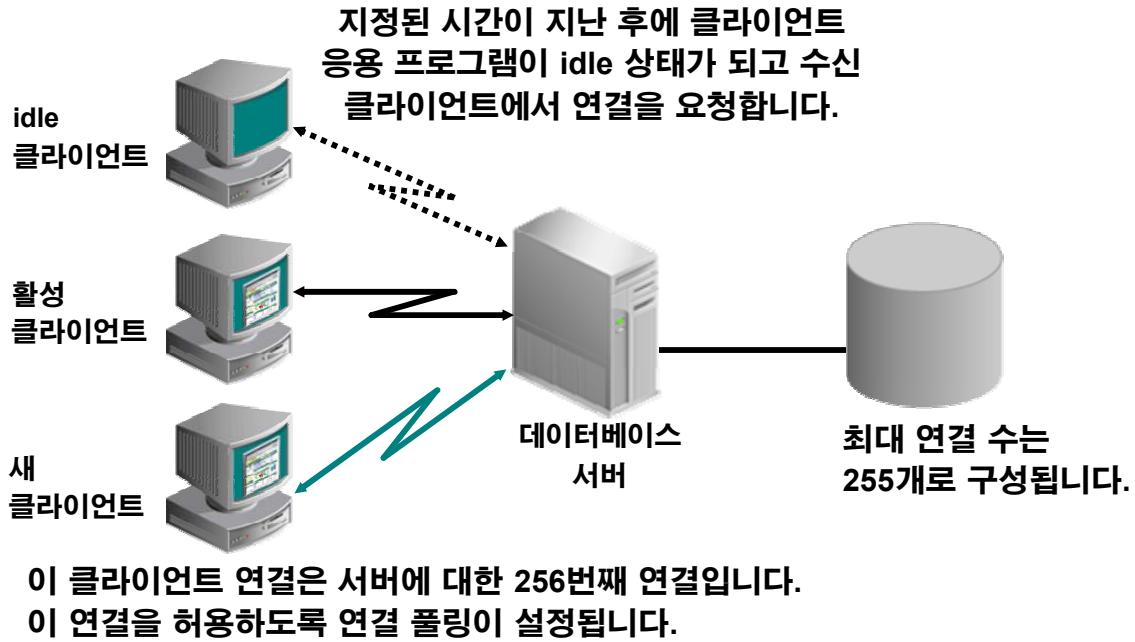
Dedicated Server를 사용하는지 또는 Shared Server를 사용하는지 여부에 따라 SGA와 PGA의 내용이 달라집니다.

- 모든 SQL 문의 텍스트와 구문이 분석된 Form은 SGA에 저장됩니다.
- Cursor State는 검색한 행과 같은 SQL 문에 대한 런타임 메모리 값을 포함합니다.
- 유저 세션 데이터는 보안과 리소스 사용 정보를 포함합니다.
- 스택 공간은 프로세스 로컬 변수를 포함합니다.

## 관련 정보

SGA와 PGA에 대한 변경 내용은 유저에게 투명하지만 다중 유저를 지원하는 경우에는 LARGE\_POOL\_SIZE 초기화 파라미터를 늘려야 합니다. 모든 서버가 모든 세션의 요청을 처리할 수 있도록 각 Shared Server 프로세스는 모든 세션의 데이터 공간에 액세스해야 합니다. SGA에는 각 세션의 데이터 공간에 대한 공간이 할당됩니다. 유저 프로파일의 General 페이지에 있는 Database Services 영역에서 PRIVATE\_SGA 리소스 한계를 설정하여 세션이 할당할 수 있는 공간을 제한할 수 있습니다.

## Shared Server: 연결 풀링



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Shared Server: 연결 풀링

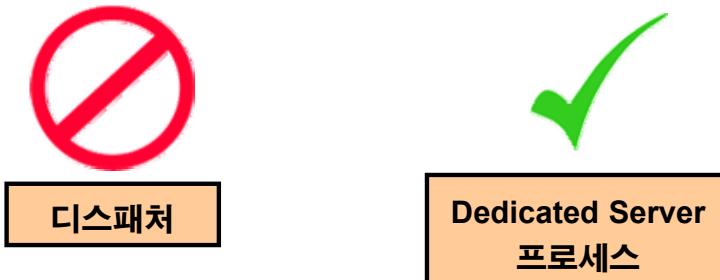
연결 풀링 기능을 사용하면 데이터베이스 서버가 idle 세션을 만료시키고 연결을 사용하여 활성 세션을 제공할 수 있습니다. idle 논리적 세션은 열린 상태로 남아있고 물리적 연결은 해당 세션에서 다음 요청이 올 때 자동으로 다시 설정됩니다. 따라서 웹 응용 프로그램이 기존 하드웨어를 사용하여 더 많은 동시 유저를 수용할 수 있습니다. 연결 풀링은 Shared Server를 통해 구성할 수 있습니다.

이 예제에서 오라클 데이터베이스 서버는 255개의 연결로 구성되며 클라이언트 중 하나는 지정된 시간이 지난 후에 idle 상태로 남습니다. 연결 풀링은 이 연결을 들어오는 클라이언트 연결(256번째 연결)에 사용할 수 있습니다. idle 클라이언트가 수행할 작업이 더 있는 경우 다른 클라이언트의 idle 연결을 사용하여 해당 클라이언트에 대해 연결이 재설정됩니다.

## Shared Server를 사용하지 않는 경우

다음은 Shared Server를 사용하여 수행하지 않아야 하는 특정 데이터베이스 작업 유형입니다.

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Shared Server를 사용하지 않는 경우

Oracle Shared Server 구조는 프로세스 및 메모리를 효율적으로 사용할 수 있는 모델이지만 모든 연결에 적합한 것은 아닙니다. Common Request Queue를 처리해야 할 뿐만 아니라 많은 유저가 디스패처 Response Queue를 공유할 수 있으므로 Shared Server는 웨어하우스 query 또는 일괄 처리와 같은 큰 데이터 집합을 다루어야 하는 작업과 함께 수행하면 제대로 작동하지 않습니다.

뒷 장에서 설명할 Oracle Recovery Manager를 사용하는 백업 및 recovery 세션에서는 매우 큰 데이터 집합도 다루며 Dedicated Server 연결을 사용해야 합니다.

Shared Server 연결을 사용하여 수행할 수 없거나 수행하지 않아야 하는 관리 작업이 많이 있습니다. 이러한 관리 작업으로는 Instance 시작 및 종료, 테이블스페이스 또는 데이터 파일 생성, 인덱스 및 테이블 유지 관리, 통계 분석 및 일반적으로 DBA가 수행하는 기타 여러 작업이 있습니다. 모든 DBA 세션은 Dedicated Server를 선택해야 합니다.

## 데이터베이스 간의 통신 구성

- 사이트 간에 데이터나 메시지를 보내려면 두 사이트 모두에 네트워크를 구성해야 합니다.
- 다음을 구성해야 합니다.
  - 네트워크 연결(예: TNSNAMES.ora)
  - 데이터베이스 링크

```
CREATE DATABASE LINK <remote_global_name>
CONNECT TO <user> IDENTIFIED BY <pwd>
USING '<connect_string_for_remote_db>';
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 간의 통신 구성

데이터베이스 링크는 다른 데이터베이스의 객체에 액세스할 수 있도록 하는 데이터베이스의 스키마 객체입니다. 다른 데이터베이스가 오라클 데이터베이스 시스템일 필요는 없습니다. 그러나 비오라클 시스템에 액세스하려면 Oracle Heterogeneous Services를 사용해야 합니다.

전용(Private) 데이터베이스 링크를 생성하려면 CREATE DATABASE LINK 시스템 권한을 가지고 있어야 합니다. 공용(Public) 데이터베이스 링크를 생성하려면 CREATE PUBLIC DATABASE LINK 시스템 권한을 가지고 있어야 합니다. 또한 원격 오라클 데이터베이스에 대한 CREATE SESSION 시스템 권한을 가지고 있어야 합니다.

응용 프로그램에서 데이터베이스 링크를 사용하여 원격 데이터베이스에 액세스하는 경우 오라클 데이터베이스는 로컬 요청을 대신하여 원격 데이터베이스에 데이터베이스 세션을 설정합니다. 원격 데이터베이스에 연결이 설정되는 방식은 데이터베이스 링크를 생성하는 데 사용되는 CONNECT TO 절에 따라 결정됩니다. 고정 유저, 현재 유저 및 연결된 유저 데이터베이스 링크를 생성할 수 있습니다. 현재 유저 링크는 Oracle Advanced Security Option을 통해서만 사용할 수 있습니다. 슬라이드의 예제는 고정 유저 데이터베이스 링크를 생성하는 구문을 보여줍니다.

데이터베이스 링크를 생성한 후에는 이를 사용하여 다른 데이터베이스의 테이블 및 뷰를 참조할 수 있습니다. SQL 문에서는 테이블 또는 뷰 이름에 @edblink를 추가하여 다른 데이터베이스의 테이블 또는 뷰를 참조할 수 있습니다. 다른 데이터베이스의 테이블 또는 뷰를 query하거나, INSERT, UPDATE, DELETE 또는 LOCK TABLE 테이블에 대해 문을 사용할 수 있습니다.

## 다른 데이터베이스에 연결

```
REMOTE_ORCL =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)  
     (HOST = host02.example.com)  
     (PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = orcl.example.com)  
    )  
  )
```

tnsnames.ora

```
CONNECT hr/hr@orcl;
```

SQL\*Plus

```
CREATE DATABASE LINK remote  
CONNECT TO HR IDENTIFIED BY HR  
USING 'REMOTE_ORCL';
```

```
SELECT * FROM employees@remote
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 다른 데이터베이스에 연결

슬라이드에서는 데이터베이스 링크를 생성하기 전에 필요한 tnsnames.ora 항목을 보여줍니다. 예제에는 연결 문자열 REMOTE\_ORCL를 사용하여 유저 HR에게 연결되는 REMOTE라는 고정 유저 데이터베이스 링크가 나와 있습니다. 데이터베이스 링크를 생성한 후에는 이를 사용하여 다른 데이터베이스의 테이블 및 뷰를 참조할 수 있습니다.

뷰에 대한 설명은 다음과 같습니다.

```
SQL> DESC DBA_DB_LINKS
```

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2 (30)
DB_LINK	NOT NULL	VARCHAR2 (128)
USERNAME		VARCHAR2 (30)
HOST		VARCHAR2 (2000)
CREATED	NOT NULL	DATE

```
SQL> select owner, db_link, username from dba_db_links;
```

OWNER	DB_LINK	USERNAME
HR	REMOTE.EXAMPLE.COM	HR

## 퀴즈

리스너를 구성하는 데 사용된 구성 파일은 무엇입니까?

1. **listener.ora**
2. **listener.conf**
3. **tnsnames.ora**
4. **tnsnames.conf**
5. **sqlnet.ora**
6. **sqlnet.conf**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1, 5

## 퀴즈

Shared server 프로세스 구조를 사용하는 경우에 PGA는 SGA로 재배치됩니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- Enterprise Manager를 사용하여 다음 작업 수행
  - 추가 리스너 생성
  - Oracle Net 서비스 alias 생성
  - Connect-time failover 구성
  - Oracle Net 리스너 제어
- tnsping을 사용하여 Oracle Net 연결 테스트
- Shared Server를 사용하는 경우와 Dedicated Server를 사용하는 경우 식별

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 6 개요: Oracle 네트워크 구성 요소 사용

이 연습에서는 다음 내용을 다룹니다.

- 다른 데이터베이스에 연결하도록 로컬 이름 분석(Names Resolution) 구성
- connect-time failover를 위한 두번째 리스너 생성

ORACLE®

Copyright © 2010, Oracle. All rights reserved.



# 데이터베이스 저장 영역 구조 관리

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

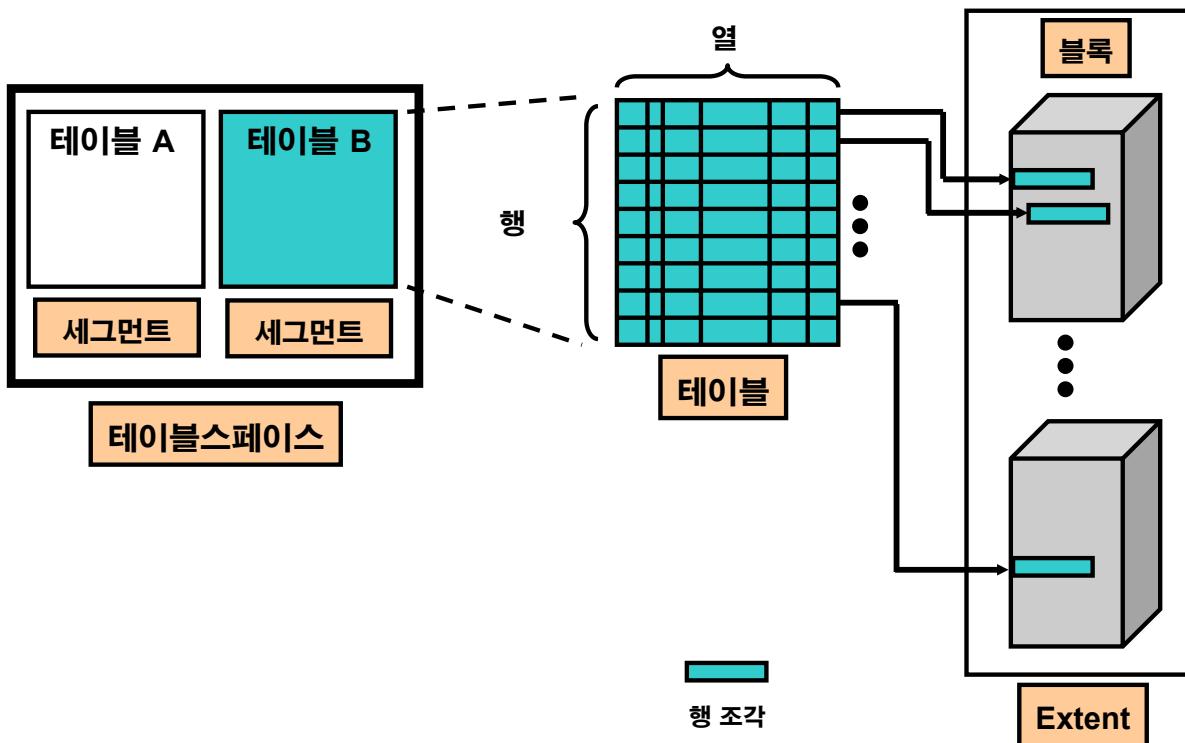
이 단원을 마치면 다음을 수행할 수 있습니다.

- 블록의 테이블 행 데이터 저장 영역 설명
- 테이블스페이스 생성 및 관리
- 테이블스페이스 정보 얻기

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 테이블 데이터가 저장되는 방법



ORACLE

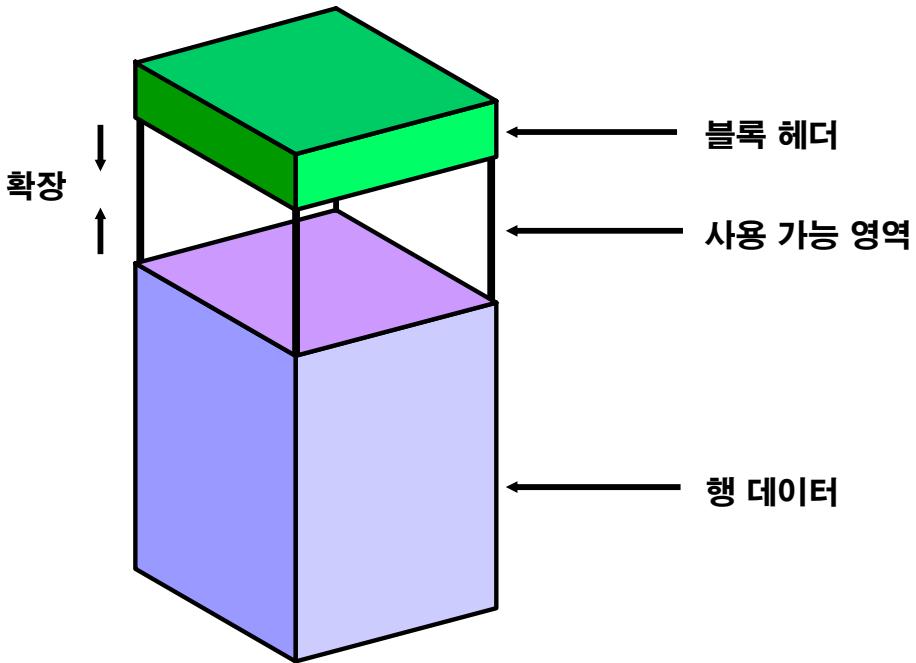
Copyright © 2010, Oracle. All rights reserved.

### 테이블 데이터가 저장되는 방법

테이블이 생성될 때 해당 테이블의 데이터를 보관하는 세그먼트가 생성됩니다. 테이블스페이스는 세그먼트 모음을 포함합니다.

논리적으로 테이블은 각 열의 값이 들어 있는 행들로 구성됩니다. 행은 행 조각의 품으로 결국 데이터베이스 블록에 저장됩니다. 행 조각이라고 하는 이유는 경우에 따라 전체 행이 한 장소에 저장되지 않을 수도 있기 때문입니다. 삽입된 행이 너무 커서 단일 블록(체인된 행)에 들어갈 수 없거나 기존 행이 갱신되어 현재 블록(이전된 행)의 사용 가능 영역보다 커지는 경우에 이러한 상황이 발생합니다. 행 조각은 또한 테이블의 열이 255개를 초과하는 경우에도 사용됩니다. 이런 경우에 조각들은 같은 블록(블록 내 체인화) 또는 다중 블록 간에 있기도 합니다.

## 데이터베이스 블록: 내용



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 블록: 내용

- **블록 헤더:** 블록 헤더에는 세그먼트 유형(테이블이나 인덱스 등), 데이터 블록 주소, 테이블 디렉토리, 행 디렉토리, 블록의 행을 수정할 때 사용되는 각각 23바이트 정도의 트랜잭션 슬롯이 포함됩니다. 블록 헤더는 맨 위부터 아래쪽으로 확장됩니다.
- **행 데이터:** 블록에 있는 행의 실제 데이터입니다. 행 데이터 공간은 맨 아래부터 위쪽으로 확장됩니다.
- **사용 가능 영역:** 사용 가능 영역은 블록의 중간에 위치하며, 필요한 경우 헤더 및 행 데이터 공간이 늘어날 수 있도록 합니다. 새 행이 삽입되거나 기존 행의 열이 더 큰 값으로 갱신되면 행 데이터는 사용 가능 영역으로 확장됩니다.

헤더 확장을 유발하는 이벤트의 예는 다음과 같습니다.

- 행 항목이 추가로 필요한 행 디렉토리
- 처음에 구성한 것보다 많은 트랜잭션 슬롯 필요

처음에는 블록의 사용 가능 영역은 연속된 공간입니다. 그러나 삭제와 갱신을 거듭하면 블록의 사용 가능 영역이 단편화될 수 있습니다. 필요한 경우 블록의 사용 가능 영역은 Oracle 서버에 의해 병합됩니다.

# 저장 영역 구조 탐색

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The top navigation bar includes 'ORACLE Enterprise Manager 11g' and 'Database Control'. Below it, the title 'Database Instance: orcl.oracle.com' is displayed. The main menu bar contains 'Home', 'Performance', 'Availability', 'Server' (which is highlighted in blue), 'Schema', and 'Data Movement'. Under the 'Server' menu, the 'Storage' section is expanded, listing various database components: Control Files, Tablespaces, Temporary Tablespace Groups, Datafiles, Rollback Segments, Redo Log Groups, Archive Logs, Disk Groups, Migrate to ASM, and Make Tablespace Locally Managed. To the right of the storage section is the 'Database Configuration' section, which includes links for Memory Advisors, Automatic Undo Management, Initialization Parameters, and View Database Feature Usage. A callout box with a black border and white background, containing the Korean text '자세한 정보를 보려면 링크를 누르십시오.' (Click the link to view detailed information.), points to the 'Storage' section.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 저장 영역 구조 탐색

논리적 데이터 구조는 데이터베이스의 물리적 파일에 저장됩니다. 데이터베이스의 논리적 구조는 EM(Enterprise Manager)을 통해 쉽게 확인할 수 있습니다. Server 페이지의 Storage 영역에 있는 링크를 누르면 각 구조에 대한 자세한 정보를 얻을 수 있습니다.

## 새 테이블스페이스 생성

**Create Tablespace**

		<input type="button" value="Show SQL"/>	<input type="button" value="Cancel"/>	<input type="button" value="OK"/>								
<input checked="" type="radio"/> General <input type="radio"/> Storage												
<p>* Name <input type="text" value="INVENTORY"/></p>												
<b>Extent Management</b>		<b>Type</b>	<b>Status</b>									
<input checked="" type="radio"/> Locally Managed <input type="radio"/> Dictionary Managed		<input checked="" type="radio"/> Permanent <input type="checkbox"/> Set as default permanent tablespace <input type="checkbox"/> Encryption <input type="button" value="Encryption Options"/>	<input checked="" type="radio"/> Read Write <input type="radio"/> Read Only <input type="radio"/> Offline									
		<input type="radio"/> Temporary <input type="checkbox"/> Set as default temporary tablespace										
		<input type="radio"/> Undo <small>Undo Retention Guarantee</small> <input type="radio"/> Yes <input checked="" type="radio"/> No										
<b>Datafiles</b>												
<input type="checkbox"/> Use bigfile tablespace <small>Tablespace can have only one datafile with no practical size limit.</small>												
<input type="button" value="Select"/> <input type="text" value="Name"/>		<input type="button" value="Directory"/>	<input type="button" value="Add"/> <input type="text" value="Size (MB)"/>									
<table border="1"> <thead> <tr> <th>Select</th> <th>Name</th> <th>Directory</th> <th>Size (MB)</th> </tr> </thead> <tbody> <tr> <td colspan="4">No items found</td> </tr> </tbody> </table>					Select	Name	Directory	Size (MB)	No items found			
Select	Name	Directory	Size (MB)									
No items found												

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 새 테이블스페이스 생성

1. Server 탭을 누른 다음 Storage 머리글 아래에서 Tablespaces를 누릅니다.
2. Create를 누릅니다.  
주: 기존 테이블스페이스와 유사한 테이블스페이스를 생성하려면 기존 테이블스페이스를 선택하고 Actions 메뉴에서 Create Like를 선택합니다. Go를 누릅니다.  
Create Tablespace 페이지가 나타납니다.
3. 테이블스페이스의 이름을 입력합니다.
4. Extent Management 머리글 아래에서 Locally Managed를 선택합니다.  
로컬관리방식 테이블스페이스의 Extent는 오라클 데이터베이스 서버에 의해 테이블스페이스 내에서 효과적으로 관리됩니다. 딕셔너리 관리방식의 테이블스페이스의 경우에는 Extent를 더 능동적으로 관리해야 하며 Extent를 추적하기 위해 데이터 딕셔너리에 액세스해야 합니다. 딕셔너리 관리 방식의 옵션은 역호환성을 위해서만 제공되므로 사용하지 않을 것을 권장합니다.
5. Type 머리글 아래에서 Permanent를 선택합니다.  
영구 테이블스페이스에는 시스템 또는 유저가 생성한 영구 데이터베이스 객체가 저장됩니다.
6. Status 머리글 아래에서 Read Write를 선택합니다.  
Read Write 상태는 테이블스페이스가 생성된 후 유저가 테이블스페이스에 대한 읽기 및 쓰기 작업을 수행할 수 있음을 의미합니다. 이것은 기본값입니다.
7. 페이지의 Datafiles 영역에서 Add를 눌러 테이블스페이스에 데이터 파일을 추가합니다.

## 새 테이블스페이스 생성

**Add Datafile**

Storage Type **Automatic Storage Management**

\* DiskGroup **DATA**

Template **<Default>**

Alias Directory

Alias Name

Tablespace **INVENTORY**

File Size **100** **MB**

Reuse Existing File

**Storage**

Automatically extend datafile when full (AUTOEXTEND)

Increment **10** **MB**

Maximum File Size  Unlimited  
 Value  **MB**

**TIP Changes made on this page will NOT take effect until you click Continue.**

**Add Datafile**

Storage Type **File System**

\* File Name

\* File Directory

Tablespace **INVENTORY**

File Size **100** **MB**

Reuse Existing File

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 새 테이블스페이스 생성(계속)

테이블스페이스에는 하나 이상의 파일이 있어야 합니다. 환경에 따라 적절한 저장 영역 유형을 선택합니다. Big File 테이블스페이스는 ASM 또는 기타 논리적 볼륨 관리자가 스트라이핑 또는 RAID(Redundant Array of Independent Disks) 및 동적으로 확장 가능한 논리적 볼륨을 지원하는 초대용량 데이터베이스에서 사용됩니다.

8. Add Datafiles 페이지에서 원하는 저장 영역 유형을 선택하고 필요한 정보를 입력합니다. ASM에서 원하는 디스크 그룹을 선택합니다. File System에 대해 데이터 파일에 대한 파일 이름 및 파일 디렉토리를 입력합니다.
9. 원하는 파일 크기를 입력합니다.
10. Storage 영역에서 "Automatically extend datafile when full (AUTOEXTEND)"를 선택한 다음 Increment 필드에 크기를 지정합니다. 이렇게 하면 공간이 부족할 때마다 데이터 파일이 자동으로 확장됩니다. 확장되는 공간은 물론 데이터 파일이 있는 물리적 매체의 크기에 의해 제한됩니다. Maximum File Size를 Unlimited로 지정하거나 최대 크기를 입력합니다. Continue를 눌러 Create Tablespace 페이지로 돌아갑니다.
12. Create Tablespace 페이지로 돌아가서 필요에 따라 Storage 탭을 눌러 이 테이블스페이스에 대한 저장 영역 옵션을 수정할 수 있습니다. 대부분의 경우 Storage 페이지에서 모든 기본값을 그대로 사용할 수 있습니다. OK를 눌러 테이블스페이스를 생성합니다.

**주:** 위 단계는 대부분의 상황에서 테이블스페이스를 빠르게 생성하는 방법을 보여줍니다. 구체적인 요구 사항에 따라 일부 옵션을 변경해야 할 수도 있습니다.

# 테이블스페이스에 대한 저장 영역

The screenshot shows the 'Storage' tab of the Oracle Database storage settings dialog. It includes sections for Extent Allocation (Automatic selected), Segment Space Management (Automatic selected), Compression Options (Disabled selected), and Enable logging (Yes selected). The Block Size is set to 8192.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 테이블스페이스에 대한 저장 영역

**Extent Allocation:** 로컬 관리 방식 테이블스페이스 내의 Extent는 다음의 두 가지 방법 중 하나로 할당할 수 있습니다.

- **Automatic:** *autoallocate*라고도 하는 이 방법을 사용하면 테이블스페이스 내의 Extent 크기가 시스템에 의해 관리됩니다. 임시 테이블스페이스에 대해서는 Automatic을 지정할 수 없습니다.
- **Uniform:** 유저가 지정한 동일 Extent 크기로 테이블스페이스가 관리됩니다. 기본 크기는 1MB입니다. 임시 테이블스페이스의 모든 Extent는 동일합니다. 언두 테이블스페이스에 대해서는 Uniform을 지정할 수 없습니다.

**Segment Space Management:** 로컬 관리 방식 테이블스페이스 내의 세그먼트 공간 관리는 다음과 같이 지정될 수 있습니다.

- **Automatic:** 오라클 데이터베이스는 비트맵을 사용하여 세그먼트 내의 사용 가능 영역을 관리합니다. 비트맵은 행 삽입에 사용할 수 있는 블록 내부 공간의 크기에 따라 세그먼트 내에 있는 각 데이터 블록의 상태를 설명합니다. 데이터 블록에서 사용 가능한 공간의 크기가 바뀌면 비트맵에 새로운 상태가 반영됩니다. 오라클 데이터베이스는 비트맵을 통해 사용 가능 영역을 보다 자동화된 방식으로 관리합니다. 따라서 이러한 공간 관리 유형을 Automatic Segment Space Management(ASSM)라고 합니다.

## 테이블스페이스에 대한 저장 영역(계속)

- **Manual:** 이 옵션을 지정하면 Free List를 사용하여 세그먼트 내의 사용 가능 영역을 관리할 수 있습니다. Free List는 행을 삽입할 공간이 있는 데이터 블록 리스트입니다. 이러한 형태의 세그먼트 공간 관리는 테이블스페이스에 생성된 스키마 객체에 대한 PCTUSED, FREELISTS 및 FREELIST GROUPS 저장 영역 파라미터를 지정 및 튜닝해야 하므로 수동 세그먼트 공간 관리라고 합니다. 이 기능은 역호환성을 위해 지원되는 기능이므로 대신 ASSM을 사용할 것을 권장합니다.

**Compression Options:** 데이터 세그먼트 압축은 기본적으로 비활성화되어 있습니다. 데이터 세그먼트 압축을 활성화하면 디스크 공간 사용을 저장하고, 버퍼 캐시의 메모리 사용을 줄이고, 읽기 작업 중 query 실행 속도를 높일 수 있습니다. 하지만 데이터 로딩 및 DML에 대한 CPU 오버헤드 비용이 듭니다. 이 방법은 특히 시간이 오래 걸리는 읽기 전용 작업이 있는 OLAP (online analytical processing) 시스템에 유용하지만 또한 OLTP(online transaction processing) 시스템에서도 사용될 수 있습니다.

압축 절을 사용하는 경우에 대한 자세한 내용은 *Oracle Database Administrator's Guide*를 참조하십시오.

**Logging:** Logging 절에서는 테이블스페이스에 생성된 세그먼트에 대한 기본 Logging 값을 설정합니다. 테이블스페이스의 객체에 대한 변경 내용은 리두 로그에 기록됩니다. Logging이 활성화되어 있지 않으면 SQL\*Loader를 사용한 모든 직접 로드 및 직접 로드 INSERT 작업이 리두 로그에 기록되지 않으므로 데이터 손실이 발생할 경우 객체를 Recovery할 수 없게 됩니다. 따라서 Logging이 활성화되지 않은 상태에서 객체를 생성하는 경우 Recovery가 가능하도록 하려면 해당 객체를 백업해야 합니다. Logging 활성화를 선택하지 않으면 이후에 객체를 Recovery 하는 기능에 상당한 영향을 줄 수 있습니다. 주의해서 사용하십시오. Logging 절에 대한 자세한 내용은 *Oracle Database SQL Reference*를 참조하십시오.

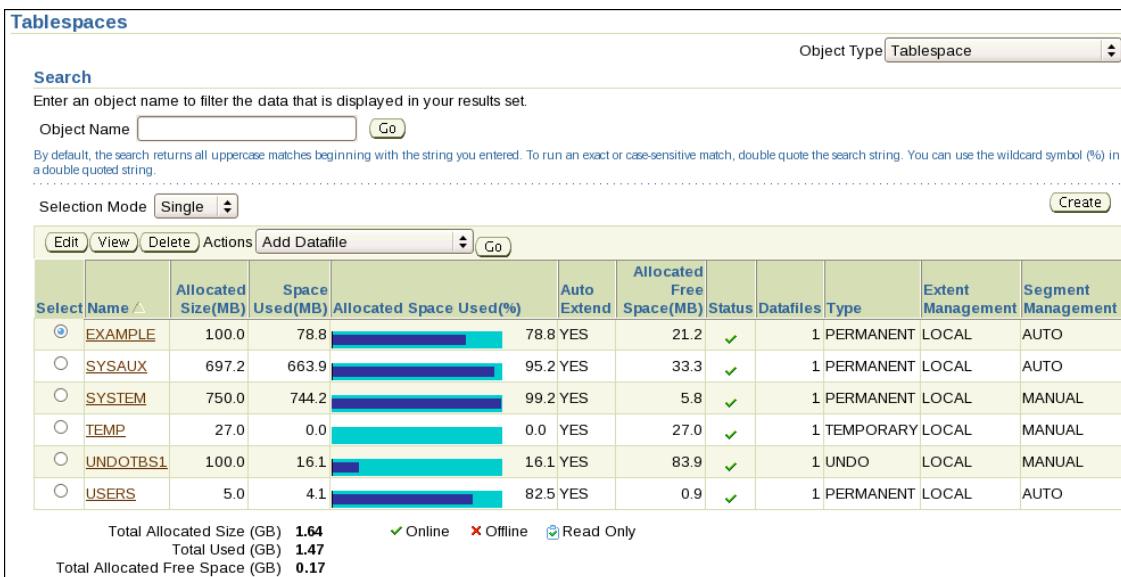
**주:** 데이터베이스가 FORCE LOGGING 모드 상태인 경우, 테이블스페이스 Logging 설정에 우선합니다. 데이터베이스는 데이터베이스 생성 시 또는 데이터베이스 생성 이후에 alter database force logging 명령을 사용하여 FORCE LOGGING 모드로 설정할 수 있습니다.

**Block Information:** 이 영역은 생성 중인 테이블스페이스에 사용되는 블록 크기를 보여줍니다. 여기에서는 이 크기가 읽기 전용 값으로 표시됩니다. 다른 블록 크기 초기화 파라미터 (DB\_nK\_CACHE\_SIZE)를 설정한 경우에는 해당 값이 여기에 옵션으로 나열됩니다.

다른 블록 크기를 정의하는 자세한 방법은 *Oracle Database Administrator's Guide*를 참조하십시오.

# 미리 구성된 데이터베이스의 테이블스페이스

- **SYSTEM**
- **SYSAUX**
- **TEMP**
- **UNDOTBS1**
- **USERS**
- **EXAMPLE (선택 사항)**



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 미리 구성된 데이터베이스의 테이블스페이스

이 과정에서는 미리 구성된 데이터베이스에 다음 테이블스페이스가 생성됩니다.

- **SYSTEM:** SYSTEM 테이블스페이스는 Oracle 서버에서 데이터베이스 관리를 위해 사용합니다. 이 테이블스페이스에는 데이터베이스에 대한 관리 정보를 포함하는 테이블 및 데이터 딕셔너리가 들어 있습니다. 이들은 모두 SYS 스키마에 포함되어 있으며 SYS 유저나 필요한 권한을 가진 관리 유저만 액세스할 수 있습니다.
  - **SYSAUX:** SYSTEM 테이블스페이스의 보조 테이블스페이스입니다. 이전 오라클 데이터베이스 릴리스 중 일부 구성 요소 및 제품에서는 SYSTEM 테이블스페이스 또는 고유 테이블스페이스를 사용했지만 이제는 SYSAUX 테이블스페이스를 사용합니다. 모든 Oracle Database 10g Release 이상의 데이터베이스에는 SYSAUX 테이블스페이스가 있어야 합니다.
- EM(Enterprise Manager)에서는 이 테이블스페이스의 내용을 파일 차트로 볼 수 있습니다. 파일 차트를 보려면 Administration 페이지에서 Tablespaces를 누릅니다. SYSAUX를 선택하고 Edit를 누릅니다. 그런 다음 Occupants 탭을 누릅니다. 생성한 후에는 EM을 사용하여 SYSAUX 테이블스페이스 내에 있는 각 보유자의 공간 사용량을 모니터할 수 있습니다. SYSAUX 테이블스페이스의 공간을 너무 많이 차지하는 구성 요소가 있거나 그럴 가능성이 있으면 보유자 중 하나를 선택하고 Change Tablespace를 눌러 보유자를 다른 테이블스페이스로 이동할 수 있습니다.

## 미리 구성된 데이터베이스의 테이블스페이스(계속)

- **TEMP:** 임시 테이블스페이스는 대규모 정렬이나 인덱스 생성 등과 같이 임시 세그먼트 생성이 필요한 SQL 문을 실행할 때 사용됩니다. 각 유저에게 생성된 데이터 객체를 저장하기 위한 기본 테이블스페이스가 할당되는 것과 마찬가지로 각 유저에게 임시 테이블스페이스가 할당됩니다. 가장 좋은 방법은 데이터베이스의 기본 임시 테이블스페이스를 정의하는 것입니다. 그러면 별도로 지정하지 않는 한 이 테이블스페이스가 모든 새로 생성된 유저에게 할당됩니다. 미리 구성된 데이터베이스에서는 TEMP 테이블스페이스가 기본 임시 테이블스페이스로 지정됩니다. 즉 유저 계정 생성 시 임시 테이블스페이스를 지정하지 않으면 오라클 데이터베이스는 이 테이블스페이스를 유저에게 할당합니다.
- **UNDOTBS1:** 데이터베이스 서버가 언두 정보를 저장하는 데 사용하는 언두 테이블스페이스입니다. 데이터베이스가 Automatic Undo Management를 사용하는 경우 지정된 시간에 단지 하나의 언두 테이블스페이스만 사용할 수 있습니다. 이 테이블스페이스는 데이터베이스가 생성될 때 생성됩니다.
- **USERS:** 이 테이블스페이스는 유저 객체 및 데이터를 저장하는 데 사용됩니다. 유저가 생성되었을 때 기본 테이블스페이스가 지정되지 않은 경우 USERS 테이블스페이스는 그 유저에 의해 생성된 모든 객체에 대한 기본 테이블스페이스입니다. SYS 및 SYSTEM 유저에게 기본 영구 테이블스페이스는 SYSTEM입니다.
- **EXAMPLE:** 이 테이블스페이스에는 데이터베이스 생성 시 설치할 수 있는 예제 스키마가 포함되어 있습니다. 이러한 예제 스키마는 예제에 사용할 수 있는 공통 플랫폼을 제공합니다. Oracle 설명서 및 교육용 소프트웨어에는 예제 스키마를 기반으로 하는 예제가 포함되어 있습니다.

주: 손쉬운 관리를 위해 별도의 인덱스용 테이블스페이스를 가지고 있는 것이 일반적입니다.

## 테이블스페이스 변경

The screenshot shows the 'Edit Tablespace' dialog for 'EXAMPLE'. The 'Extent Management' section is highlighted with a red box. Inside, 'Locally Managed' is selected. An arrow points from this section to the 'Status' section, where 'ReadWrite' is also selected. Other tabs like 'General' and 'Storage' are visible.

Select Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/> EXAMPLE	100.0	78.8	78.8	YES	21.2	<span style="color: green;">✓</span>	1	PERMANENT LOCAL	AUTO	
<input type="radio"/> SYSAUX										

**Edit Tablespace: EXAMPLE**

**General** **Storage** **Thresholds**

Name: EXAMPLE  
Bigfile tablespace: No

**Extent Management**

- Locally Managed
- Dictionary Managed

**Type**

- Permanent
  - Set as default permanent tablespace
  - Encryption ([Encryption Options](#))
- Temporary
  - Set as default temporary tablespace
- Undo

**Status**

- Read Write
- Read Only
- Offline

Offline Mode: Normal

**Datafiles**

Select Name	Directory	Size (MB)	Used (MB)
<input checked="" type="radio"/> example_265_688820635	+DATA/orcl/datafile/	100.00	78.81

Add

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 테이블스페이스 변경

테이블스페이스를 생성한 후 시스템 요구 사항이 변경되면 여러 가지 방법으로 테이블스페이스를 변경할 수 있습니다.

**이름 변경:** 테이블스페이스의 새 이름을 입력한 후 Apply를 누릅니다.

**상태 변경:** 테이블스페이스의 상태에는 세 가지가 있습니다. 테이블스페이스 유형에 따라 다음 세 가지 상태 중 일부를 사용하지 못할 수도 있습니다.

- **Read Write:** 테이블스페이스가 온라인이며 읽기 및 쓰기 작업을 수행할 수 있습니다.
- **Read Only:** Read Only를 지정하면 테이블스페이스가 변환 Read Only 모드로 설정됩니다. 이 상태에서 기존 트랜잭션을 완료(커밋 또는 롤백)할 수는 있지만 테이블스페이스의 객체에 대해 더 이상 DML(데이터 조작어) 작업을 수행할 수 없습니다. 테이블스페이스는 Read Only 상태에서 온라인입니다. SYSTEM 또는 SYSAUX 테이블스페이스를 읽기 전용으로 지정할 수는 없습니다.

**주:** 언두 및 임시 테이블스페이스는 읽기 전용으로 지정될 수 없습니다.

## 테이블스페이스 변경(계속)

- **Offline:** 온라인 테이블스페이스를 오프라인으로 설정하여 임시로 해당 부분의 데이터베이스를 일반적인 용도로 사용할 수 없도록 할 수 있습니다. 이 부분 이외의 테이블스페이스는 열려 있어 유저가 데이터에 액세스할 수 있습니다. 오프라인 상태로 설정할 때 다음 옵션을 사용할 수 있습니다.
  - **Normal:** 테이블스페이스의 모든 데이터 파일에 오류 조건이 없는 경우 정상적으로 테이블스페이스를 오프라인으로 설정할 수 있습니다. 오라클 데이터베이스는 테이블스페이스의 모든 데이터 파일을 오프라인으로 설정할 경우 해당 파일에 대해 체크포인트를 수행하는 방식으로 모든 데이터가 디스크에 기록되도록 합니다.
  - **Temporary:** 테이블스페이스의 하나 이상의 파일에 오류 조건이 있어도 임시로 테이블스페이스를 오프라인으로 설정할 수 있습니다. 오라클 데이터베이스는 아직 오프라인이 아닌 데이터 파일을 오프라인으로 설정하고 체크포인트를 수행합니다. 오프라인 파일이 없지만 임시 절을 사용하는 경우 테이블스페이스를 다시 온라인으로 되돌리는 데는 Media Recovery가 필요하지 않습니다. 단, 테이블스페이스에서 하나 이상의 파일이 쓰기 오류로 인해 오프라인이며 유저가 테이블스페이스를 오프라인으로 임시 설정하는 경우 해당 테이블스페이스를 다시 온라인으로 되돌리려면 recovery가 필요합니다.
  - **Immediate:** 오라클 데이터베이스에서 데이터 파일의 체크포인트를 수행하지 않고 테이블스페이스를 즉시 오프라인으로 설정할 수 있습니다. Immediate를 지정한 경우 테이블스페이스를 온라인으로 되돌리려면 Media Recovery가 필요합니다. 데이터베이스가 NOARCHIVELOG 모드로 실행되고 있는 경우에는 테이블스페이스를 즉시 오프라인으로 설정할 수 없습니다.
  - **For Recover:** FOR RECOVER 설정은 지원이 중단되었습니다. 이 구문은 역호환성을 위해서만 지원됩니다.

**주:** 시스템 테이블스페이스는 오프라인으로 설정할 수 없습니다.

**Changing the size:** 테이블스페이스에 데이터 파일을 추가하거나 기존 데이터 파일의 크기를 변경하여 기존 테이블스페이스에 공간을 추가할 수 있습니다.

- 테이블스페이스에 새 데이터 파일을 추가하려면 Add를 누른 다음 Add Datafile 페이지에 데이터 파일에 대한 정보를 입력합니다.
- 주:** Big File 테이블스페이스에 추가 데이터 파일을 추가할 수는 없습니다.
- 기존 데이터 파일의 크기를 변경하려면 Edit Tablespace 페이지의 Datafiles 영역에서 데이터 파일 이름을 눌러 데이터 파일을 선택하거나, 데이터 파일을 선택하고 Edit를 누릅니다. 그런 다음 Edit Datafile 페이지에서 데이터 파일의 크기를 변경할 수 있습니다. 테이블스페이스를 확장하거나 축소할 수 있지만 데이터 파일을 파일에서 사용된 공간보다 작게 만들 수 없습니다. 이렇게 하려고 시도하면 다음과 같은 오류 메시지가 표시됩니다.  
`ORA-03297: file contains used data beyond requested  
RESIZE value`

**Storage options:** Storage를 눌러 테이블스페이스의 logging 방식을 변경합니다.

**Thresholds:** Thresholds를 눌러 테이블스페이스에서 warning 또는 critical 레벨의 공간 사용에 도달하는 시점을 변경합니다. 다음 세 가지 옵션이 있습니다.

- **Use Database Default Thresholds:** 사전 설정된 기본값을 사용합니다. 이 기본값을 설정하는 옵션이 제공됩니다.
- **Specify Thresholds:** 이 옵션을 선택하면 해당 테이블스페이스에 대한 임계값을 설정할 수 있습니다.
- **Disable Thresholds:** 이 옵션은 해당 테이블스페이스에 대한 공간 사용 Alert를 해제합니다.

**주:** 공간 활용률은 기본적으로 매 10분마다 확인되며, 임계값 alert 등록에 몇 분이 걸릴 수도 있습니다.

## 테이블스페이스에서 가능한 작업

The screenshot shows the Oracle Database 11g Tablespace Management interface. A red arrow points from the 'Actions' menu in the top navigation bar down to the 'Generate DDL' option in the dropdown menu. The 'Generate DDL' option is highlighted in blue. Below the menu, a table displays information about various tablespaces, including their names, sizes, and management details. At the bottom of the interface, there is a 'Show DDL' section containing the generated DDL code.

Select	Name	Allocated Size(MB)	Used (%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/>	EXAMPLE	100.0	78.8	YES	21.2	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSAUX	697.2	95.2	YES	33.3	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSTEM	750.0	99.2	YES	5.8	✓	1	PERMANENT	LOCAL	MANUAL
<input type="radio"/>	TEMP	27.0	0.0	YES	27.0	✓	1	TEMPORARY	LOCAL	MANUAL
<input type="radio"/>	UNDOTBS1	100.0	17.1	YES	82.9	✓	1	UNDO	LOCAL	MANUAL
<input type="radio"/>	USERS	5.0	4.1		0.9	✓	1	PERMANENT	LOCAL	AUTO

Total Allocated Size (GB) **1.64**      ✓ Online    ✘ Offline    ⚡ Read Only  
 Total Used (GB) **1.47**  
 Total Allocated Free Space (GB) **0.17**

Show DDL

```
CREATE SMALLFILE TABLESPACE "EXAMPLE" DATAFILE '+DATA/orcl/datafile/example.265.688820635'
SIZE 100M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M NOLOGGING EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 테이블스페이스에서 가능한 작업

Actions 메뉴를 사용하여 테이블스페이스에서 다양한 작업을 수행할 수 있습니다.

테이블스페이스를 선택한 다음 수행할 작업을 선택합니다.

- **Add Datafile:** 테이블스페이스에 데이터 파일을 추가하여 테이블스페이스를 확장합니다.
- **Create Like:** 테이블스페이스를 템플릿으로 사용하여 다른 테이블스페이스를 생성합니다.
- **Generate DDL:** 테이블스페이스를 생성하는 DDL(데이터 정의어) 문을 생성합니다. 생성한 DDL 문을 복사하여 텍스트 파일에 붙여 넣은 다음 스크립트로 사용하거나 설명서에서 사용할 수 있습니다.
- **Make Locally Managed:** 테이블스페이스가 현재 딕셔너리 관리 방식인 경우 로컬관리방식으로 변환합니다. 이 변환은 한쪽 방향으로만 가능합니다. 즉 테이블스페이스를 다시 딕셔너리 관리 방식으로 변환할 수는 없습니다. 필요한 경우 PL/SQL 패키지인 DBMS\_SPACE\_ADMIN.TABLESPACE\_MIGRATE\_FROM\_LOCAL을 사용하여 딕셔너리 관리 방식으로 변환할 수도 있습니다.
- **Make Readonly:** 테이블스페이스에 대한 모든 쓰기 작업을 정지합니다. 현재 트랜잭션의 쓰기 작업은 완료할 수 있지만 테이블스페이스에서 새 DML 또는 다른 쓰기 작업을 시작할 수 없습니다. 테이블스페이스가 현재 읽기 전용이 아닌 경우에만 이 작업이 표시됩니다.
- **Make Writable:** 테이블스페이스의 객체에 DML 및 다른 쓰기 작업을 허용합니다. 테이블스페이스가 현재 쓰기 가능 상태가 아닌 경우에만 이 작업이 표시됩니다.

## 테이블스페이스에서 가능한 작업(계속)

- **Place Online:** 현재 오프라인 상태인 테이블스페이스를 온라인 상태로 만듭니다.
- **Reorganize:** Reorganization Wizard를 시작합니다. 이 마법사를 통해 테이블스페이스 내에서 객체를 이동하여 사용되지 않는 공간을 확보할 수 있습니다. 이 작업은 테이블스페이스 객체의 사용량이 적은 시간에 수행되어야 합니다.
- **Run Segment Advisor:** Segment Advisor를 시작합니다. Segment Advisor를 사용하면 객체 내의 공간 단편화 레벨을 기반으로 객체에 사용 가능한 공간이 있는지 확인할 수 있습니다. 테이블스페이스 레벨에서 테이블스페이스의 모든 세그먼트에 대해 권고가 생성됩니다.
- **Show Dependencies:** 해당 테이블스페이스가 종속되어 있는 객체나 해당 테이블스페이스에 종속된 객체를 표시합니다.
- **Show Tablespace Contents:** 모든 Extent의 그래픽 맵을 포함하여 테이블스페이스의 모든 세그먼트에 대한 정보를 보여줍니다.
- **Take Offline:** 현재 온라인 상태인 테이블스페이스를 사용할 수 없게 만듭니다. 테이블스페이스가 삭제되는 것은 아니며 단지 사용할 수 없는 상태가 됩니다.

# 테이블스페이스 삭제

## Warning

Once a tablespace has been dropped, the objects and data in it will no longer be available. To recover them can be a time consuming process. Oracle recommends a backup before and after dropping a tablespace.

Are you sure you want to delete Tablespace EXAMPLE?

Delete associated datafiles from storage

Select	Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/>	EXAMPLE	100.0	78.8	78.8 YES		21.2	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSAUX	697.2	663.9	95.2 YES		33.3	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSTEM	750.0	744.2	99.2 YES		5.8	✓	1	PERMANENT	LOCAL	MANUAL
<input type="radio"/>	TEMP	27.0	0.0	0.0 YES		27.0	✓	1	TEMPORARY	LOCAL	MANUAL
<input type="radio"/>	UNDOTBS1	100.0	17.1	17.1 YES		82.9	✓	1	UNDO	LOCAL	MANUAL
<input type="radio"/>	USERS	5.0	4.1	82.5 YES		0.9	✓	1	PERMANENT	LOCAL	AUTO

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 테이블스페이스 삭제

더 이상 필요하지 않은 테이블스페이스와 해당 컨텐트(테이블스페이스에 포함된 세그먼트)를 데이터베이스에서 삭제할 수 있습니다. 테이블스페이스를 삭제하려면 DROP TABLESPACE 시스템 권한이 있어야 합니다.

테이블스페이스를 삭제하면 연관된 데이터베이스의 콘트롤 파일에 있는 파일 포인터가 제거됩니다. OMF(Oracle 관리 파일)를 사용하는 경우에는 기본 운영 체제 파일도 제거됩니다. OMF를 사용하지 않는 경우 삭제된 테이블스페이스를 구성하는 운영 체제 파일(데이터 파일)을 삭제하도록 지정할 수도 있습니다. 테이블스페이스를 삭제할 때 Oracle 서버에서 데이터 파일도 함께 삭제하도록 지정하지 않은 경우 나중에 데이터 파일을 삭제하려면 운영 체제의 해당 명령을 사용해야 합니다.

활성 세그먼트가 포함된 테이블스페이스는 삭제할 수 없습니다. 예를 들어, 테이블스페이스에 있는 테이블이 현재 사용 중이거나 테이블스페이스에 커밋되지 않은 트랜잭션을 롤백하는 데 필요한 언두 데이터가 포함되어 있는 경우에는 해당 테이블스페이스를 삭제할 수 없습니다. 테이블스페이스는 온라인이어도 되고 오프라인이어도 되지만 삭제하기 전에 오프라인으로 설정하는 것이 좋습니다.

## 테이블스페이스 정보 보기

```
SELECT tablespace_name, status, contents, logging, extent_management,  
allocation_type, segment_space_management  
FROM dba_tablespaces
```

TABLESPACE_NAME	STATUS	CONTENTS	LOGGING	EXTENT_MANAGEMENT	ALLOCATION_TYPE	SEGMENT_SPACE_MANAGEMENT
SYSTEM	ONLINE	PERMANENT	LOGGING	LOCAL	SYSTEM	MANUAL
SYSAUX	ONLINE	PERMANENT	LOGGING	LOCAL	SYSTEM	AUTO
UNDOTBS1	ONLINE	UNDO	LOGGING	LOCAL	SYSTEM	MANUAL
TEMP	ONLINE	TEMPORARY	NOLOGGING	LOCAL	UNIFORM	MANUAL
USERS	ONLINE	PERMANENT	LOGGING	LOCAL	SYSTEM	AUTO
EXAMPLE	ONLINE	PERMANENT	NOLOGGING	LOCAL	SYSTEM	AUTO

```
SELECT ts#, name FROM v$tablespace
```

TS#	NAME
0	SYSTEM
1	SYSAUX
2	UNDOTBS1
4	USERS
3	TEMP
6	EXAMPLE

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 테이블스페이스 정보 보기

선택된 테이블스페이스에 대한 정보를 보려면 View를 누릅니다. View Tablespace 페이지에서 Edit를 눌러 테이블스페이스를 변경할 수도 있습니다.

다음을 query하여 테이블스페이스 및 데이터 파일에 대한 정보를 얻을 수도 있습니다.

- 테이블스페이스 정보:
  - DBA\_TABLESPACES
  - V\$TABLESPACE
- 데이터 파일 정보:
  - DBA\_DATA\_FILES
  - V\$DATAFILE

주: V\$DBFILE 뷰에는 데이터베이스의 모든 데이터 파일이 표시됩니다. 이 뷰는 기록 호환성을 위해 보존됩니다. 대신 V\$DATAFILE을 사용하는 것이 권장됩니다.

- 임시 파일 정보:
  - DBA\_TEMP\_FILES
  - V\$TEMPFILE

# 테이블스페이스 내용 보기

**Show Tablespace Contents**

Size (MB)	100.0	Used (MB)	78.8	Extent Mgmt	LOCAL	Auto Extend	Yes
Block Size (KB)	8	Used (%)	78.8	Segment Mgmt	AUTO	Extents	882

**Segments**

Search

Segment Name	Type	Minimum Size (KB)	Minimum Extents
SH.CUSTOMERS	All Types		
PM.SYS_LOB0000			
PM.SYS_LOB0000			
SH.SUPPLEMENTAL			
OE.PRODUCT_DES			
SH.SALES.SALES_			
SH.SALES.SALES_			
SH.SALES.SALES_			
SH.CUSTOMERS_F			
SH.SALES.SALES_			

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

**Extent Map**

Clicking the Highlight Extents button for a segment in the table will cause all extents that belong to that segment to be highlighted in the Extent Map. Clicking on a used extent in the Extent Map will select the segment to which that extent belongs in the segment table.

Header  
Used  
Free  
Selected  
Unmapped

Zoom 75%

**Extent Map**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 테이블스페이스 내용 보기

기본 테이블스페이스 페이지 또는 특정 테이블스페이스 페이지에서, Actions drop-down list에서 Show Tablespace Contents를 선택한 다음 Go를 누릅니다. Show Tablespace Contents 페이지에는 테이블스페이스의 세그먼트 리스트, 각 세그먼트의 유형, 세그먼트 크기 및 각 세그먼트의 Extent 개수 등 테이블스페이스에 대한 상세한 정보가 표시됩니다. 열 헤더를 눌러 리스트를 정렬하거나 Search 영역에 값을 입력하여 리스트를 필터링할 때 이 네 가지 값을 사용할 수 있습니다. 덕셔너리 관리방식의 테이블스페이스인 경우 다음과 같은 추가 열이 표시됩니다.

- Max Extents
- Next
- Percent Increase

Extent list를 보려면 Extents 열의 링크를 누릅니다.

그래픽 형식으로 Extent를 보려면 "Extent 맵"을 확장하고 커서를 개별 Extent 위로 이동합니다. 다음 정보가 표시됩니다.

- Extent가 속한 세그먼트 이름
- Extent ID
- 블록 ID
- Extent 크기(블록 단위)
- Extent가 저장된 데이터 파일

# OMF(Oracle 관리 파일)

파일 이름이 아닌 데이터베이스 객체 관점에서 파일 작업을 지정합니다.

파라미터	설명
DB_CREATE_FILE_DEST	데이터 파일 및 임시 파일에 대한 기본 파일 시스템 디렉토리의 위치를 정의합니다.
DB_CREATE_ONLINE_LOG_DEST_n	리두 로그 파일 및 콘트롤 파일 생성을 위한 위치를 정의합니다.
DB_RECOVERY_FILE_DEST	Fast Recovery Area의 기본 위치입니다.

예제:

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '+DATA';
SQL> CREATE TABLESPACE tbs_1;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## OMF(Oracle 관리 파일)

Oracle 관리 파일을 사용하면 오라클 데이터베이스를 구성하는 운영 체제 파일을 직접 관리하지 않아도 됩니다. 파일 이름이 아닌 데이터베이스 객체 관점에서 작업을 지정합니다. 그러면 데이터베이스는 내부적으로 표준 파일 시스템 인터페이스를 사용하여 다음 데이터베이스 구조에 대한 파일을 필요에 따라 생성하고 삭제합니다.

- 테이블스페이스
- 리두 로그 파일
- 콘트롤 파일
- 아카이브된 로그
- 블록 변경 사항 추적 파일
- Flashback 로그
- RMAN 백업

데이터베이스에는 Oracle 관리 파일과 비관리 파일이 섞여 있을 수 있습니다. 데이터베이스에서 디렉토리를 직접 생성하지 않으므로 이러한 파라미터 중 하나로 지정된 파일 시스템 디렉토리가 존재해야 합니다. 또한 디렉토리에는 데이터베이스에서 해당 디렉토리에 파일을 생성하도록 허용하는 권한이 있어야 합니다.

위의 예제는 DB\_CREATE\_FILE\_DEST가 설정된 후에는 CREATE TABLESPACE 문에서 DATAFILE 절이 생략될 수 있음을 보여줍니다. 데이터 파일은 DB\_CREATE\_FILE\_DEST에 의해 지정된 위치에 생성됩니다. 예제에서처럼 테이블스페이스를 생성하면 모든 파라미터에 기본값이 할당됩니다.

## OMF(Oracle 관리 파일)(계속)

Oracle 관리 파일은 고유의 이름 지정 형식이 있습니다. 예를 들어, Linux 및 Unix 기반 시스템에서는 다음 형식을 사용합니다.

```
<destination_prefix>/o1_mf_%t_%u_.dbf
```

Oracle 관리 파일 이름은 변경하지 마십시오. 데이터베이스는 Oracle 관리 파일을 이름으로 식별합니다. 파일 이름을 변경하는 경우 데이터베이스는 더 이상 Oracle 관리 파일로 인식할 수 없으며, 파일을 올바르게 관리할 수 없습니다.

다음 예제는 데이터 파일 생성 위치를 /u01/oradata로 설정하고 나서 이 위치에 데이터 파일이 포함된 tbs\_1 테이블스페이스를 생성합니다.

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/u01/oradata';
SQL> CREATE TABLESPACE tbs_1;
```

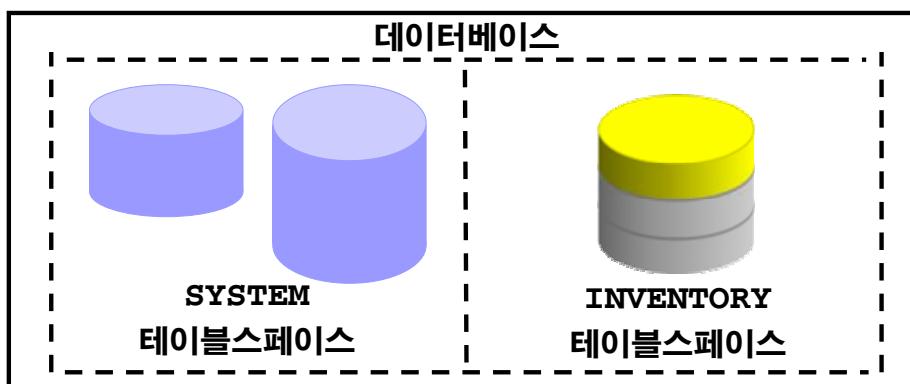
기본적으로 SYSTEM 및 SYSAUX 테이블스페이스를 포함한 Oracle 관리 데이터 파일은 100MB이며 자동 확장됩니다.

**주:** 기본적으로 ASM은 OMF 파일을 사용하지만, 테이블스페이스 생성 시 또는 기존 테이블스페이스에 ASM 데이터 파일을 추가할 때 ASM 데이터 파일에 대한 alias 이름을 지정한 경우에는 해당 파일이 OMF가 될 수 없습니다.

## 데이터베이스 확장

다음과 같은 방법으로 데이터베이스를 확장할 수 있습니다.

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



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 확장

Enterprise Manager 또는 SQL 문을 사용하여 이 작업을 수행할 수 있습니다. 데이터베이스의 크기는 모든 테이블스페이스 크기의 합계로 설명될 수 있습니다.

## 퀴즈

데이터베이스에는 Oracle 관리 파일과 비관리 파일이 섞여 있을 수 있습니다.

1. 참
2. 거짓

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1

## 퀴즈

Big File 테이블스페이스에는 적어도 100MB의 파일이  
한 개 있어야 합니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 블록의 테이블 행 데이터 저장 영역 설명
- 테이블스페이스 생성 및 관리
- 테이블스페이스 정보 얻기

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 7 개요: 데이터베이스 저장 영역 구조 관리

이 연습에서는 다음 내용을 다룹니다.

- 테이블스페이스 생성
- 테이블스페이스에 대한 정보 수집



Copyright © 2010, Oracle. All rights reserved.



# 8 유저 보안 관리

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 데이터베이스 유저 계정 생성 및 관리:
  - 유저 인증
  - 기본 저장 영역(테이블스페이스) 할당
- 권한 부여 및 취소
- 룰 생성 및 관리
- 프로파일 생성 및 관리:
  - 표준 암호 보안 기능 구현
  - 유저별 리소스 사용량 제어

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

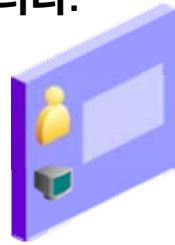
다음은 데이터베이스 유저를 관리하고 목표를 이해하는 데 도움이 되는 관련 용어입니다.

- 데이터베이스 유저 계정(*Database User Account*)은 데이터베이스 객체에 대한 소유권 및 액세스 권한을 구성하는 수단입니다.
- 암호(*password*)는 오라클 데이터베이스에 의한 인증입니다.
- 권한(*privilege*)은 특정 유형의 SQL 문을 실행하거나 다른 유저의 객체에 액세스할 수 있는 권한입니다.
- 룰(*Role*)은 유저나 다른 룰에 부여된 관련 권한의 명명된 그룹입니다.
- 프로파일(*Profile*)은 데이터베이스 사용 및 Instance 리소스에 명명된 리소스 제한 집합을 적용하고 계정 상태 및 암호 관리 규칙을 관리합니다.
- 할당량(*Quota*)은 주어진 테이블스페이스에 허용된 공간입니다. 할당량은 유저의 리소스 사용을 제어할 수 있는 방법 중 하나입니다.

# 데이터베이스 유저 계정

각 데이터베이스 유저 계정에는 다음이 포함되어 있습니다.

- 고유 Username
- 인증 방식
- 기본 테이블스페이스
- 임시 테이블스페이스
- 유저 프로파일
- 초기 Consumer Group
- 계정 상태



## 스키마:

- 데이터베이스 유저가 소유하는 데이터베이스 객체의 모음
- 유저 계정과 동일한 이름을 가짐

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 유저 계정

데이터베이스에 액세스하려면 유저가 유효한 데이터베이스 유저 계정을 지정하고 이 유저 계정에 필요한 인증에 성공해야 합니다. 각 데이터베이스 유저에게는 고유한 데이터베이스 계정이 있습니다. 오라클은 잠재적인 보안 취약성을 방지하고 특정 감사(audit) 작업에 대해 의미 있는 데이터를 제공하기 위해 고유한 데이터베이스 계정을 사용할 것을 권장합니다. 그러나 유저가 일반 데이터베이스 계정을 공유하는 경우가 있습니다. 이 경우에 운영 체제와 응용 프로그램은 데이터베이스에 대해 적절한 보안을 제공해야 합니다. 각 유저 계정에는 다음이 포함됩니다.

- **고유 Username:** Username은 30바이트를 초과하거나 특수 문자를 포함할 수 없으며 문자로 시작되어야 합니다.
- **인증 방식:** 가장 일반적인 인증 방식은 암호이지만, Oracle Database 11g에서는 Password, Global 및 External 인증 방식(예: 생물 통계학적(Biometrics) 인증, 인증서 및 토큰 인증)을 지원합니다.
- **기본 테이블스페이스:** 유저가 다른 테이블스페이스를 지정하지 않은 경우 객체를 생성하는 장소입니다. 기본 테이블스페이스가 있다고 해서 유저에게 이 테이블스페이스에 객체를 생성할 수 있는 권한이 있거나 기본 테이블스페이스 내에 객체를 생성할 공간 할당량이 있는 것은 아니라는 점에 유의하십시오. 이는 모두 별도로 부여됩니다.
- **임시 테이블스페이스:** Instance가 정렬 및 임시 테이블 등의 임시 객체를 유저 대신 생성하는 위치입니다. 임시 테이블스페이스에는 할당량이 적용되지 않습니다.

## 데이터베이스 유저 계정(계속)

- **유저 프로파일:** 유저에게 할당되는 리소스 및 암호 제한 집합입니다.
- **초기 Consumer Group:** 리소스 관리자가 사용합니다.
- **계정 상태:** 유저는 "열린" 계정에만 액세스할 수 있습니다. account\_status는 "locked" 상태와 "expired" 상태가 다양하게 조합된 형태일 수 있습니다.

**스키마:** 스키마는 데이터베이스 유저가 소유하는 데이터베이스 객체의 모음입니다. 스키마 객체는 데이터베이스의 데이터를 직접 참조하는 논리적 구조입니다. 스키마 객체에는 테이블, 뷰, 시퀀스, 내장 프로시저, 동의어, 인덱스, 클러스터 및 데이터베이스 링크가 있습니다.

일반적으로 스키마 객체에는 응용 프로그램이 데이터베이스에 생성하는 모든 것이 포함됩니다.

**주:** 데이터베이스 유저가 반드시 사람일 필요는 없습니다. 즉, 일반적으로 HR과 같은 특정 응용 프로그램의 데이터베이스 객체를 소유하는 유저를 생성하게 됩니다. 데이터베이스 유저는 장치나 응용 프로그램일 수도 있고 보안용으로 데이터베이스 객체를 그룹화하는 방식일 수도 있습니다. 데이터베이스 유저에게는 특정인의 개인 식별 정보가 필요하지 않습니다.

## 미리 정의된 관리 계정

- **SYS 계정:**
  - DBA 를 및 기타 몇 가지 룰이 부여됨
  - ADMIN OPTION과 함께 모든 권한을 가짐
  - 시작, 종료 및 일부 유지 관리 명령에 필요
  - 데이터 딕셔너리 및 AWR(Automatic Workload Repository) 소유
- **SYSTEM 계정은 DBA, MGMT\_USER 및 AQ\_ADMINISTRATOR\_ROLE 룰이 부여됨**
- **DBSNMP 계정은 OEM\_MONITOR 룰이 부여됨**
- **SYSMAN 계정은 MGMT\_USER, RESOURCE 및 SELECT\_CATALOG\_ROLE 룰이 부여됨**
- **이 계정들은 일상적인 작업에는 사용되지 않습니다.**

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 미리 정의된 관리 계정

SYS 및 SYSTEM 계정에는 기본적으로 데이터베이스 관리자(DBA) 룰이 부여됩니다. 또한 SYS 계정은 ADMIN OPTION에 대한 모든 권한을 포함하며 데이터 딕셔너리를 소유합니다. SYS 계정에 연결하려면 데이터베이스 Instance에 대해 AS SYSDBA 절을 사용해야 하며 자동 저장 영역 관리(ASM) Instance에 대해서는 AS SYSASM을 사용해야 합니다. SYSDBA 권한이 부여된 모든 유저는 AS SYSDBA 절을 사용하여 SYS 계정에 연결할 수 있습니다. SYSDBA, SYSOPER 또는 SYSASM 권한이 부여된 "권한 있는" 유저만이 Instance를 시작 및 종료 할 수 있습니다. SYSTEM 계정에는 SYSDBA 권한이 없습니다. 또한 SYSTEM에는 AQ\_ADMINISTRATOR\_ROLE 및 MGMT\_USER 룰이 부여됩니다. SYS 및 SYSTEM 계정은 데이터베이스에서 필수 계정이며 삭제할 수 없습니다.

DBSNMP 계정은 Enterprise Manager의 관리 에이전트가 데이터베이스를 모니터 및 관리하는 데 사용됩니다. SYSMAN 계정은 Oracle Enterprise Manager 관리 작업을 수행하는 데 사용됩니다. DBSNMP 또는 SYSMAN은 SYSDBA 권한이 없습니다.

**최적의 사용법(Best Practice Tip):** 이러한 계정은 최소 권한 원칙이 적용되며 일상적인 작업에는 사용되지 않습니다. DBA 권한이 필요한 유저는 부여된 필수 권한과 함께 별도의 권한을 가집니다. 예를 들어, Jim에게는 jim이라는 낮은 권한의 계정과 jim\_dba라는 높은 권한의 계정이 부여됩니다. 이 방법을 사용하면 최소 권한 원칙을 적용할 수 있고 계정을 공유할 필요가 없으며 개별 작업을 감사(audit)할 수 있습니다.

# 유저 생성

Database Instance: orcl.oracle.com > Users > Create User

Logged in As SYS

**Create User**

**General** Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

\* Name: mydba  
Profile: DEFAULT  
Authentication: Password  
\* Enter Password: .....  
\* Confirm Password: .....

For Password choice, the role is authorized via password.  
 Expire Password now

Default Tablespace: USERS  
Temporary Tablespace: TEMP

Status:  Locked  Unlocked

Show SQL

```
CREATE USER "MYDBA" PROFILE "DEFAULT" IDENTIFIED BY "*****" DEFAULT
TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK
GRANT "CONNECT" TO "MYDBA"
```

OK Cancel Show SQL Return

**Server > Users를 선택한 다음 Create 버튼을 누릅니다.**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 유저 생성

Enterprise Manager의 Users 페이지에서는 현재 데이터베이스에 액세스하도록 허용되는 데이터베이스 유저를 관리할 수 있습니다. 이 페이지를 사용하여 유저를 생성, 삭제하고 설정을 수정할 수 있습니다.

데이터베이스 유저를 생성하려면 다음을 수행하십시오.

- Enterprise Manager Database Control에서 Server 탭을 누른 후에 Security 섹션에서 Users를 누릅니다.
- Create 버튼을 누릅니다.

필요한 정보를 입력합니다. Name과 같은 필수 항목에는 별표(\*)가 표시되어 있습니다.

지정된 이름은 데이터베이스 객체를 생성하는 데 사용되는 것과 동일한 규칙을 따라야 합니다. 이 단원의 다음 페이지에서는 인증에 대한 자세한 정보를 제공합니다. 프로파일은 이 단원 뒷부분에서 다룹니다.

각 유저에게 기본 테이블스페이스 및 임시 테이블스페이스를 할당합니다. 객체를 생성할 때 유저가 테이블스페이스를 지정하지 않으면 객체는 객체 소유자에게 할당된 기본 테이블스페이스에 생성됩니다. 그러므로 객체가 생성되는 위치를 제어할 수 있습니다. 기본 테이블스페이스를 선택하지 않으면 시스템에 정의된 기본 영구 테이블스페이스가 사용됩니다. 이는 임시 테이블스페이스에도 마찬가지인데, 테이블스페이스를 지정하지 않으면 시스템에 정의된 임시 테이블스페이스가 사용됩니다.

**주:** 지원되는 SQL 구문을 보려면 Show SQL을 누릅니다. 유저 생성을 위한 전체 SQL 구문은 *Oracle® Database SQL Language Reference* 설명서를 참조하십시오.

# 유저 인증

- **Password**
- **External**
- **Global**

**Edit User: HR**

Actions: Create Like, Go, Show SQL, Revert, Apply

General Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

Name: HR  
Profile: DEFAULT  
Authentication: Password

\* Enter Password: (dropdown menu showing Password, External, Global)  
\* Confirm Password:

For Password choice, the role is authorized via password.  
 Expire Password now

Default Tablespace: USERS  
Temporary Tablespace: TEMP

Status:  Locked  Unlocked

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 유저 인증

인증은 데이터, 리소스 또는 응용 프로그램을 사용하려는 주체(유저, 장치 또는 기타 엔티티)의 ID를 검증하는 작업을 의미합니다. ID를 검증하면 원활한 상호 작용을 위한 신뢰 관계가 수립됩니다. 또한 인증을 통해 액세스 권한 및 작업을 특정 ID와 연결할 수 있으므로 책임 소재를 명백히 구분할 수 있습니다. 인증 이후에는 권한 부여 프로세스를 통해 특정 엔티티에 허용된 작업 및 액세스 권한 레벨을 허용하거나 제한할 수 있습니다.

유저를 생성할 때는 이 유저에 사용할 인증 기법을 결정해야 합니다. 이때 결정한 인증 기법은 나중에 수정할 수 있습니다.

**Password:** 오라클 데이터베이스에 의한 인증을 의미합니다. 유저가 연결을 시도할 때 입력해야 하는 연관된 암호와 함께 각 유저를 생성합니다. 암호 설정 시에는 해당 암호가 즉시 만료되도록 할 수 있으며 이렇게 하면 유저는 첫 로그인 후에 암호를 변경해야 합니다. 유저 암호를 만료시키려면 먼저 유저가 암호를 변경할 수 있는지 여부를 확인합니다. 일부 응용 프로그램에는 이러한 기능이 없습니다. Oracle Database 11g에서 생성한 모든 암호는 기본적으로 대소문자를 구분하며, 멀티바이트 문자를 포함할 수 있고 30바이트로 제한됩니다. Oracle Database 11g로 업그레이드된 데이터베이스에서 생성한 각 암호는 변경될 때까지 대소문자를 구분하지 않습니다. 암호는 네트워크를 통해 전송하기 전에 네트워크(클라이언트/서버 및 서버/서버) 연결 중에 AES(Advanced Encryption Standard) 알고리즘을 사용하여 항상 자동으로 투명하게 암호화(encryption)됩니다.

## 유저 인증(계속)

**External:** 데이터베이스 외부의 방법(운영 체제, Kerberos 또는 Radius)을 사용하는 인증입니다. Kerberos 또는 Radius의 경우에는 Advanced Security Option이 필요합니다. Username이나 암호를 지정하지 않고 오라클 데이터베이스에 쉽게 연결할 수 있습니다. 강력한 인증 방법인 Advanced Security Option을 사용하면 생물 통계학적(Biometrics) 인증, x509 인증 및 토큰 장치를 사용하여 유저를 식별할 수 있습니다. External 인증을 사용하면 데이터베이스는 기본 운영 체제, 네트워크 인증 서비스 또는 외부 인증 서비스를 사용하여 데이터베이스 계정에 대한 액세스를 제한합니다. 데이터베이스 암호는 이러한 로그인 유형에 사용되지 않습니다. 운영 체제 또는 네트워크 서비스에서 허용하는 경우에는 데이터베이스 암호를 사용하여 유저를 인증할 수 있습니다.

운영 체제 인증을 사용하려면 OS\_AUTHENT\_PREFIX 초기화 파라미터를 설정하고 이 접두어를 Oracle Username에 사용합니다. OS\_AUTHENT\_PREFIX 파라미터는 오라클 데이터베이스가 각 유저의 운영 체제 계정 이름 앞에 추가하는 접두어를 정의합니다. Oracle 소프트웨어의 역호환성을 위해 이 파라미터의 기본값은 OPS\$입니다. 유저가 연결을 시도할 때 오라클 데이터베이스는 접두어가 있는 Username을 데이터베이스에 있는 Oracle Username과 비교합니다. 예를 들어, OS\_AUTHENT\_PREFIX가 아래와 같이 설정되었다고 가정합니다.

OS\_AUTHENT\_PREFIX=OPS\$

운영 체제 계정 이름이 tsmith인 유저가 오라클 데이터베이스에 연결하여 운영 체제에 의해 인증 받는 경우 오라클 데이터베이스는 해당 데이터베이스 유저인 OPS\$tsmith가 있는지 확인하고 이 유저가 있는 경우 유저의 연결을 허용합니다. 운영 체제에서 인증되는 유저에 대한 모든 참조는 OPS\$tsmith와 같이 접두어를 포함해야 합니다.

**주:** 일부 운영 체제에서는 OS\_AUTHENT\_PREFIX 초기화 파라미터 텍스트의 대소문자를 구분합니다. 이 초기화 파라미터에 대한 자세한 내용을 보려면 사용 중인 운영 체제와 관련된 오라클 설명서를 참조하십시오.

**Global:** Oracle Advanced Security Option으로 Global 인증을 사용하면 Oracle Internet Directory를 사용하여 유저를 식별할 수 있습니다.

고급 인증 방식에 대한 자세한 사항은 *Oracle Database Security* 과정을 참조하십시오.

# 관리자 인증

## 운영 체제 보안:

- DBA는 파일을 생성하고 삭제하기 위한 OS 권한을 가져야 합니다.
- 일반 데이터베이스 유저는 데이터베이스 파일을 생성 또는 삭제할 수 있는 OS 권한을 가질 수 없습니다.

## 관리자 보안:

- **SYSDBA, SYSOPER 및 SYSASM 연결:**
  - Password file 및 강력한 인증 방식에 대해 DBA 유저가 이름별로 감사(audit)됩니다.
  - OS 인증에 대해 OS 계정 이름이 감사됩니다.
  - 권한이 있는 유저에 대해서는 OS 인증이 Password file 인증보다 우선합니다.
  - Password file은 대소문자를 구분하는 암호를 사용합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 관리자 인증

**운영 체제 보안:** UNIX 및 Linux에서 DBA는 기본적으로 데이터베이스 파일을 생성 및 삭제하는데 필요한 권한을 가진 oinstall OS 그룹에 속합니다.

**관리자 보안:** 권한이 있는 SYSDBA, SYSOPER 및 SYSASM 유저에 대한 연결은 Password file 또는 OS 권한을 사용한 검증을 수행한 이후에만 인증됩니다. OS 인증이 사용되는 경우 데이터베이스는 제공되는 Username과 암호를 사용하지 않습니다. OS 인증은 Password file이 없거나, 제공된 Username 또는 암호가 해당 파일에 없거나, Username과 암호가 제공되지 않는 경우에 사용됩니다. Oracle Database 11g의 Password file은 기본적으로 대소문자를 구분하는 암호를 사용합니다.

그러나 Password file을 통한 인증에 성공하면 해당 Username으로 연결이 기록됩니다.

운영 체제를 통한 인증에 성공하면 특정 유저를 기록하지 않는 CONNECT / 연결이 적용됩니다.

**주:** 운영 체제의 OSDBA 또는 OSOPER 그룹 멤버이고, SYSDBA 또는 SYSOPER로 연결하는 경우에는 지정한 Username 및 암호에 관계없이 관련된 관리 권한으로 연결됩니다. SYSASM의 경우 Username 또는 암호(예: sqlplus / as SYSASM)를 지정하지 않아야 합니다.

Oracle Database 11g에서 권한이 있는 유저는 Advanced Security Option에 대한 사용권이 있는 경우 Kerberos, SSL 또는 디렉토리 인증 등의 강력한 인증 방식을 사용할 수 있습니다.

## 유저 계정 Lock 해제 및 암호 재설정

The screenshot shows the Oracle Database 11g User Management interface. In the Actions column of the user list, the 'Unlock User' option is highlighted with a red box and a mouse cursor. A yellow callout box contains the Korean text: "유저를 선택하고 Unlock User를 선택하고 Go를 누릅니다." (Select the user and choose 'Unlock User', then click 'Go'). Above the list, the 'Actions' dropdown menu is open, showing options like 'Lock User', 'Unlock User', and 'Go'. The 'Go' button is also highlighted with a red box.

Select	UserName	Status	Create Date	Default Tablespace	Temporary Tablespace	Profile	Created
<input checked="" type="radio"/>	ANONYMOUS	EXPIRE LOCKED	10:51 Aug 4, 2008	SYSAUX	TEMP	DEFAULT	Aug 3, 2007 1:34:38 AM MDT
<input checked="" type="radio"/>	APEX PUBLIC USER	EXPIRED & LOCKED	7:10:51 PM MDT	USERS	TEMP	DEFAULT	Aug 3, 2007 2:04:08 AM MDT
<input checked="" type="radio"/>	BI	EXPIRED & LOCKED	7:10:51 PM MDT	USERS	TEMP	DEFAULT	Aug 4, 2008 7:04:49 PM MDT

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 유저 계정 Lock 해제 및 암호 재설정

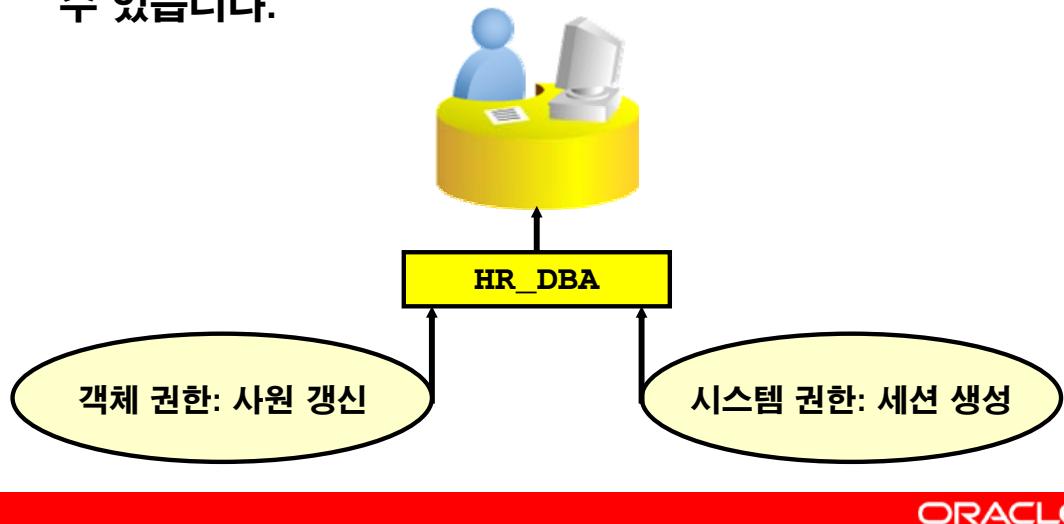
설치 및 데이터베이스 생성 시 Oracle에서 제공하는 많은 데이터베이스 유저 계정의 lock을 해제하고 재설정할 수 있습니다. 이 때 유저 계정의 lock을 해제하지 않은 경우 나중에 Users 페이지에서 유저를 선택하고 Actions list에서 Unlock User를 선택한 다음 Go를 눌러서 유저의 Lock을 해제할 수 있습니다. 이렇게 해도 암호는 변경되지 않습니다. 유저의 Lock을 해제했을 때 암호가 만료된 경우 유저를 편집하고 암호를 변경할 때까지 암호가 만료된 상태로 유지됩니다. 유저의 lock을 해제하고 암호를 재설정하려면 Edit Users 페이지에서 다음 단계를 수행합니다.

1. Enter Password 및 Confirm Password 필드에 새 암호를 입력합니다.
2. Unlocked 체크 박스를 선택합니다.
3. Apply를 눌러 암호를 재설정하고 유저 계정의 lock을 해제합니다.

# 권한

다음과 같은 두 가지 유형의 유저 권한이 있습니다.

- 시스템 권한: 유저가 데이터베이스에서 특정 작업을 수행할 수 있도록 합니다.
- 객체 권한: 유저가 특정 객체를 액세스 및 조작할 수 있습니다.



Copyright © 2010, Oracle. All rights reserved.

## 권한

권한(privilege)은 특정 유형의 SQL 문을 실행하거나 다른 유저의 객체에 액세스할 수 있는 권한입니다. 오라클 데이터베이스를 사용하면 데이터베이스 내에서 유저가 수행할 수 있거나 수행할 수 없는 작업을 제어할 수 있습니다.

권한은 두 개의 범주로 나뉩니다.

- 시스템 권한: 유저는 각 시스템 권한을 사용하여 특정 데이터베이스 작업이나 특정 유형의 데이터베이스 작업을 수행할 수 있습니다. 예를 들어, 테이블스페이스 생성 권한은 시스템 권한입니다. 시스템 권한은 관리자나 권한을 관리할 수 있는 권한을 명시적으로 받은 유저가 부여할 수 있습니다. 170가지 이상의 시스템 권한이 있습니다. 많은 시스템 권한에는 ANY 절이 포함됩니다.
- 객체 권한: 유저는 객체 권한을 사용하여 테이블, 뷰, 시퀀스, 프로시저, 함수 또는 패키지 등의 특정 객체에 대해 특정 작업을 수행할 수 있습니다. 권한이 없는 유저는 자신이 소유한 객체에만 액세스할 수 있습니다. 객체 권한은 객체의 소유자, 관리자 또는 객체에 대한 권한을 부여할 수 있는 권한을 명시적으로 받은 유저가 부여할 수 있습니다.

# 시스템 권한

The screenshot shows the 'Edit User: HR' interface. The 'System Privileges' tab is selected. On the right, there is a grid titled 'System Privilege' with columns for privilege names and 'Admin Option' checkboxes. A red box highlights the 'Edit List' button at the top right of this grid. Below the grid, a modal dialog titled 'Modify System Privileges' is open. It contains two lists: 'Available System Privileges' (containing various system privileges like ALTER SESSION, CREATE DATABASE LINK, etc.) and 'Selected System Privileges' (containing the privileges selected from the available list). A red arrow points from the 'Available System Privileges' list to the 'Selected System Privileges' list.

## 시스템 권한

시스템 권한을 부여하려면 Edit User 페이지에서 Systems Privileges 탭을 누릅니다. 사용 가능한 권한 리스트에서 알맞은 권한을 선택하고 Move 화살표를 눌러 Selected System Privileges list로 이동합니다.

ANY 절을 사용하면 해당 스키마의 범위를 벗어나 권한을 부여할 수 있습니다. 예를 들어, CREATE TABLE 권한이 있는 경우에는 테이블을 자신의 스키마 내에만 생성할 수 있습니다. SELECT ANY TABLE 권한은 다른 유저가 소유한 테이블에서 선택할 수 있습니다. SYS 유저와 DBA 룰을 소유한 유저에게는 모든 ANY 권한이 부여되므로 모든 데이터 객체에 대해 원하는 작업을 모두 수행할 수 있습니다. ANY 시스템 권한의 범위는 Oracle Database Vault Option을 사용하여 제어할 수 있습니다.

Admin Option 체크 박스를 선택하면 해당 유저가 권한을 관리하고 시스템 권한을 다른 유저에게 부여할 수 있습니다.

시스템 권한을 부여하기 위한 SQL 구문은 다음과 같습니다.

```
GRANT <system_privilege> TO <grantee clause> [WITH ADMIN OPTION]
```

시스템 권한을 부여하기 전에 보안 요구 사항을 주의 깊게 고려해야 합니다. 일부 시스템 권한은 일반적으로 관리자에게만 부여됩니다.

- **RESTRICTED SESSION:** 이 권한을 사용하면 데이터베이스가 제한 모드로 열려 있어도 로그인할 수 있습니다.

## 시스템 권한(계속)

- **SYSDBA 및 SYSOPER:** 이 권한을 사용하면 데이터베이스에서 종료, 시작, recovery 및 다른 관리 작업을 수행할 수 있습니다. SYSOPER는 유저가 기본적인 작업을 수행할 수 있지만 유저 데이터를 볼 수 있는 권한은 없습니다. 다음 시스템 권한을 포함합니다.
  - STARTUP 및 SHUTDOWN
  - CREATE SPFILE
  - ALTER DATABASE OPEN/MOUNT/BACKUP
  - ALTER DATABASE ARCHIVELOG
  - ALTER DATABASE RECOVER (Complete Recovery의 경우만 해당됩니다. UNTIL TIME|CHANGE|CANCEL|CONTROLFILE과 같은 모든 유형의 Incomplete Recovery에는 SYSDBA로 연결해야 함)
  - RESTRICTED SESSION
- SYSDBA 시스템 권한에는 Incomplete Recovery 및 데이터베이스 삭제 권한이 추가로 부여됩니다. SYSDBA 시스템 권한을 통해 유저는 SYS 유저로 연결할 수 있습니다.
- **SYSASM:** 이 권한을 사용하면 ASM Instance를 시작, 종료 및 관리할 수 있습니다.
- **DROP ANY object:** DROP ANY 권한을 사용하면 다른 스키마 유저가 소유한 객체를 삭제할 수 있습니다.
- **CREATE, MANAGE, DROP 및 ALTER TABLESPACE:** 이러한 권한을 사용하면 테이블스페이스 속성 생성, 삭제 및 변경과 같은 테이블스페이스 관리를 수행할 수 있습니다.
- **CREATE LIBRARY:** 오라클 데이터베이스에서는 개발자가 PL/SQL 내에서 C 라이브러리와 같은 외부 코드를 생성 및 호출할 수 있습니다. 라이브러리의 이름은 데이터베이스의 LIBRARY 객체가 지정해야 합니다. CREATE LIBRARY 권한을 사용하면 유저가 PL/SQL에서 실행할 수 있는 임의 코드 라이브러리를 생성할 수 있습니다.
- **CREATE ANY DIRECTORY:** 보안을 위해서 코드가 있는 운영 체제 디렉토리는 가상 오라클 디렉토리 객체에 연결되어야 합니다. CREATE ANY DIRECTORY 권한을 사용하면 비보안 코드 객체를 호출할 수 있습니다. CREATE ANY DIRECTORY 권한을 사용하면 Oracle 소프트웨어 소유자가 액세스할 수 있는 임의의 디렉토리에 유저가 읽기 및 쓰기 권한이 있는 디렉토리 객체를 생성할 수 있습니다. 즉, 유저가 해당 디렉토리의 외부 프로시저에 액세스할 수 있습니다. 유저가 데이터 파일, 리두 로그 및 감사(audit) 로그와 같은 데이터베이스 파일에 대해 직접 읽기/쓰기를 시도할 수 있습니다. 조직에 이와 같은 강력한 권한의 남용을 방지하는 보안 전략이 있어야 합니다.
- **GRANT ANY OBJECT PRIVILEGE:** 이 권한은 자신이 소유하지 않은 객체에 대해 객체 사용 권한을 부여할 수 있습니다.
- **ALTER DATABASE 및 ALTER SYSTEM:** 매우 강력한 권한으로, 데이터 파일 이름 변경 또는 버퍼 캐시 비우기와 같은 데이터베이스 및 Oracle Instance 수정 작업을 수행할 수 있습니다.

# 객체 권한

**1** Select Object Type: Table

**2** Available Privileges: SELECT

**3** Selected Privileges: SELECT

**객체를 검색하고 선택합니다.**

**객체 권한을 부여하려면 다음을 수행하십시오.**

- 객체 유형을 선택합니다.
- 객체를 선택합니다.
- 권한을 선택합니다.

## 객체 권한

객체 권한을 부여하려면 Edit User 페이지에서 Object Privileges 탭을 누릅니다. 권한을 부여할 객체 유형을 선택하고 Add 버튼을 누릅니다. <username.object name>을 입력하거나 리스트에서 선택하는 방식으로 객체를 선택합니다.

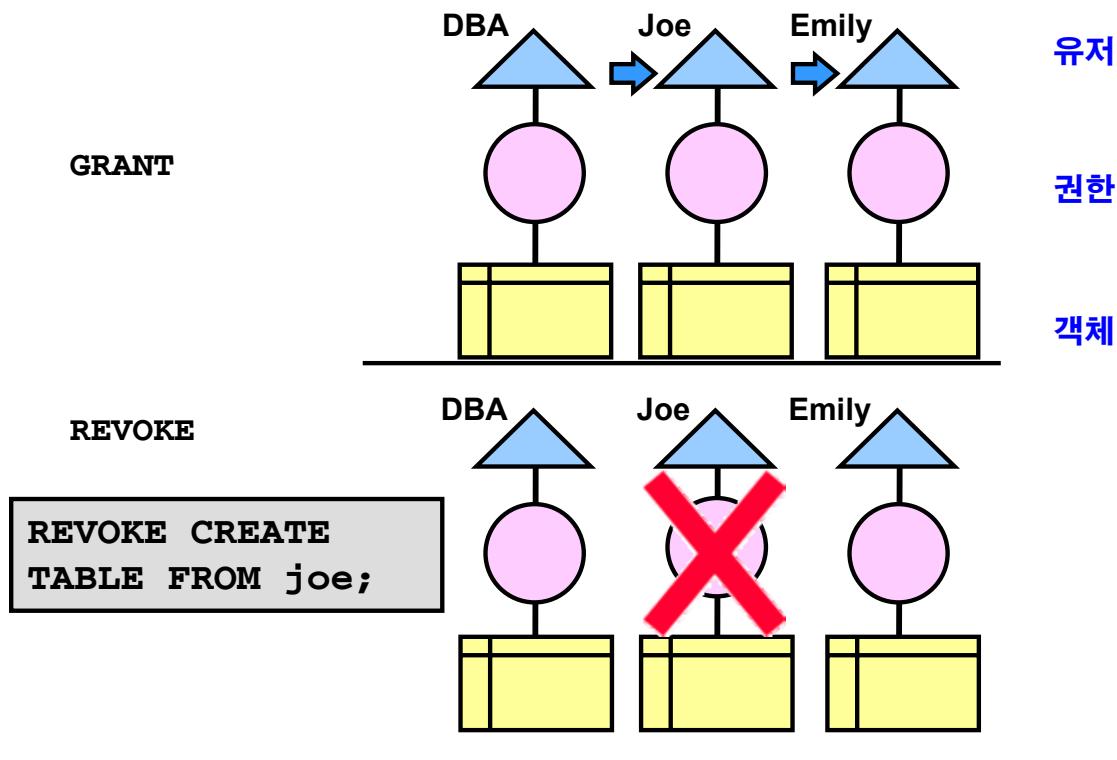
그린 다음 Available Privileges list에서 적합한 권한을 선택하고 Move 버튼을 누릅니다. 권한 선택을 마친 후에 OK를 누릅니다.

Edit User 페이지에서 이 유저가 다른 유저에게 동일한 액세스 권한을 부여할 수 있는 경우 Grant 체크 박스를 선택합니다.

객체 권한을 부여하기 위한 SQL 구문은 다음과 같습니다.

```
GRANT <object_privilege> ON <object> TO <grantee_clause>
[WITH GRANT OPTION]
```

## ADMIN OPTION을 사용하여 시스템 권한 취소



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### ADMIN OPTION을 사용하여 시스템 권한 취소

GRANT 명령으로 직접 부여된 시스템 권한은 REVOKE SQL 문을 사용하여 취소할 수 있습니다. 시스템 권한에 대해 ADMIN OPTION을 가진 유저는 다른 데이터베이스 유저의 권한을 취소할 수 있습니다. 이 작업은 원래 권한을 부여한 유저가 아니어도 가능합니다.

ADMIN OPTION이 부여되었는지 여부에 관계없이 시스템 권한을 취소할 때는 연쇄적으로 취소되지 않습니다.

시스템 권한을 취소하기 위한 SQL 구문은 다음과 같습니다.

```
REVOKE <system_privilege> FROM <grantee_clause>
```

이 슬라이드에서는 다음과 같은 상황을 보여줍니다.

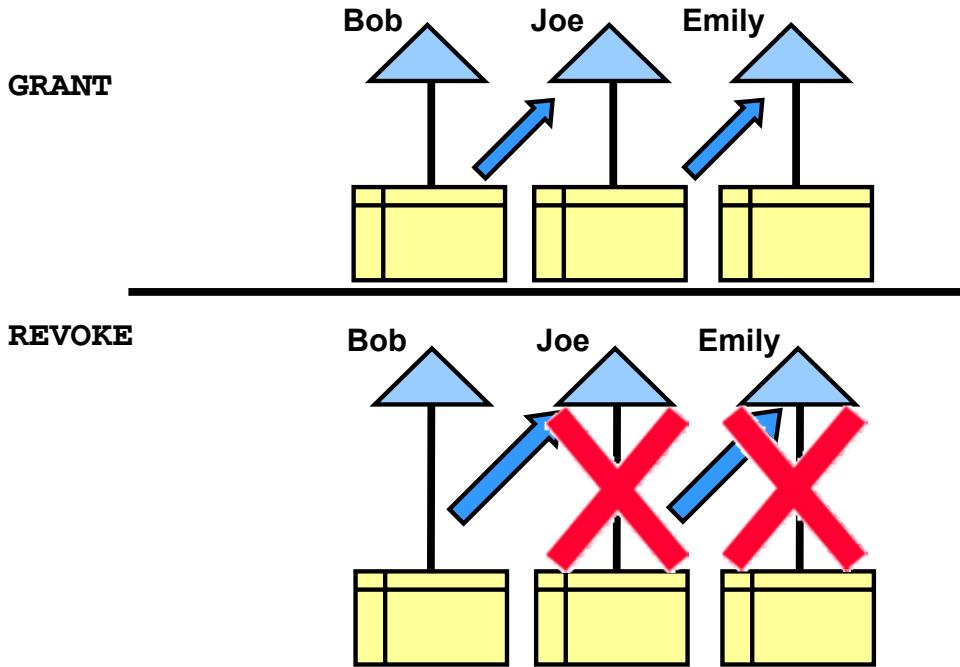
#### 시나리오

1. DBA가 ADMIN OPTION을 사용하여 CREATE TABLE 시스템 권한을 Joe에게 부여합니다.
2. Joe가 테이블을 생성합니다.
3. Joe가 CREATE TABLE 시스템 권한을 Emily에게 부여합니다.
4. Emily가 테이블을 생성합니다.
5. DBA가 Joe에게 부여된 CREATE TABLE 시스템 권한을 취소합니다.

#### 결과

Joe의 테이블은 여전히 존재하지만 Joe는 새 테이블을 생성할 수 없습니다. Emily의 테이블은 여전히 존재하며 Emily는 CREATE TABLE 시스템 권한을 갖습니다.

## GRANT OPTION을 사용하여 객체 권한 취소



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### GRANT OPTION을 사용하여 객체 권한 취소

DML(데이터 조작어) 작업과 관련된 시스템 권한을 취소할 때 연쇄적인 결과를 관찰할 수 있습니다. 예를 들어, 유저에게 SELECT ANY TABLE 권한이 부여되고 해당 유저가 테이블을 사용하는 프로시저를 생성한 경우 유저의 스키마에 포함된 모든 프로시저를 다시 사용하려면 재컴파일해야 합니다.

객체 권한 취소는 GRANT OPTION이 제공되는 경우에도 연쇄적인 결과를 나타냅니다. 유저는 자신이 부여한 권한만 취소할 수 있습니다. 예를 들어, Bob은 Joe가 Emily에게 부여한 객체 권한을 취소할 수 없습니다. 권한을 부여 받은 사람 또는 GRANT ANY OBJECT PRIVILEGE라고 부르는 권한이 있는 유저만 객체 권한을 취소할 수 있습니다.

#### 시나리오

1. GRANT OPTION을 사용하여 EMPLOYEES에 대한 SELECT 객체 권한이 Joe에게 부여됩니다.
2. Joe가 EMPLOYEES에 대한 SELECT 권한을 Emily에게 부여합니다.
3. Joe에게서 SELECT 권한이 취소됩니다. 이 취소는 Emily에게도 영향을 미칩니다.

## 롤 사용 시의 이점

- 권한 관리 용이성
- 동적 권한 관리
- 권한의 선택적 가용성



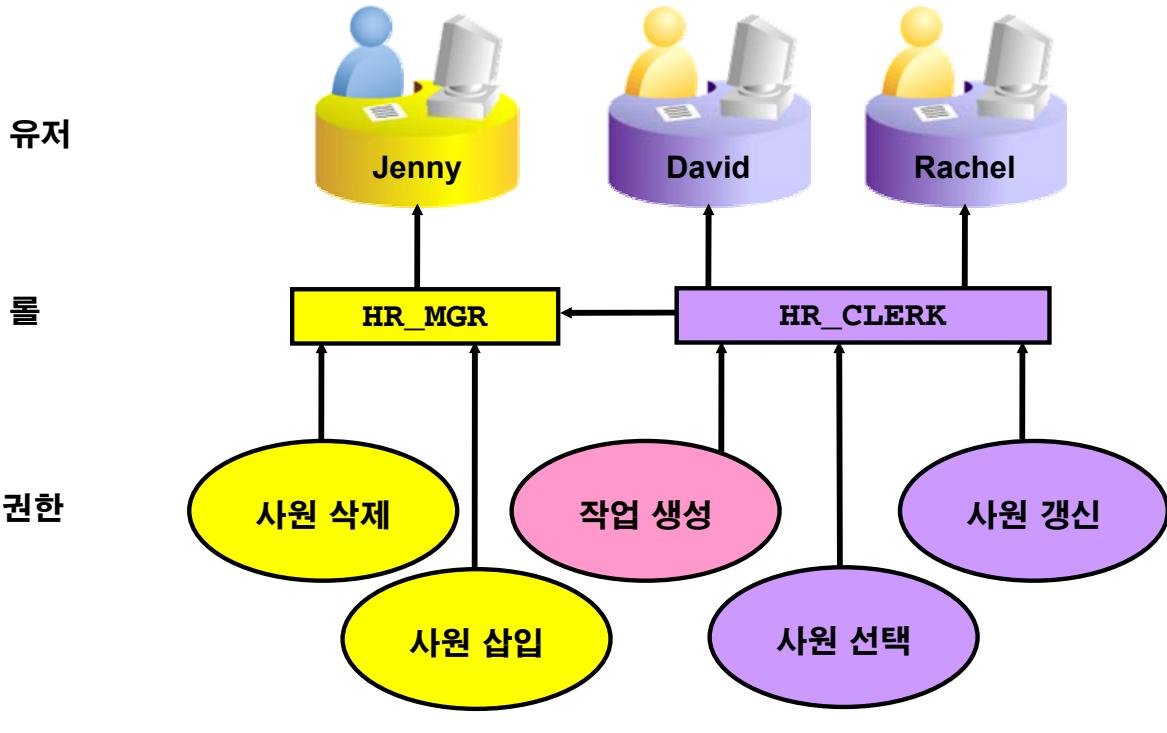
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 롤 사용 시의 이점

- **권한 관리 용이성:** 룰을 사용하여 권한 관리를 단순화합니다. 여러 유저에게 일련의 동일한 권한을 부여하는 대신 권한을 룰에 부여한 다음 이 룰을 각 유저에게 부여할 수 있습니다.
- **동적 권한 관리:** 룰과 연관된 권한을 수정하면 그 룰을 부여 받은 모든 유저는 자동으로 즉시 수정된 권한을 얻게 됩니다.
- **권한의 선택적 가용성:** 룰을 활성화 및 비활성화하여 권한을 일시적으로 설정 및 해제할 수 있습니다. 그러면 특정 상황에서 유저의 권한을 제어할 수 있습니다.

## 롤에 권한 할당 및 유저에게 룰 할당



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 롤에 권한 할당 및 유저에게 룰 할당

대부분의 시스템에서는 각 유저에 대해 개별적으로 필요한 권한을 부여하려면 시간이 많이 걸리고 오류가 발생하기 쉽습니다. Oracle 소프트웨어는 룰을 통해 권한 관리를 쉽게 제어할 수 있습니다. 룰은 유저나 다른 룰에 부여된 관련 권한의 명명된 그룹이며 데이터베이스의 권한 관리를 용이하게 하여 보안을 향상시키도록 고안되었습니다.

#### 룰 특성

- 유저와 마찬가지로 룰에도 권한을 부여하고 취소할 수 있습니다.
- 시스템 권한과 마찬가지로 룰을 유저나 다른 룰에 부여하거나 취소할 수 있습니다.
- 시스템 및 객체 권한으로 구성될 수 있습니다.
- 룰을 부여 받은 각 유저에 대해 활성화 및 비활성화될 수 있습니다.
- 룰을 활성화하려면 암호가 필요할 수 있습니다.
- 룰은 유저가 소유하지 않으며 스키마에 존재하지 않습니다.

슬라이드 예제에서 employees 테이블에 대한 SELECT 및 UPDATE 권한과 CREATE JOB 시스템 권한은 HR\_CLERK 룰에 부여됩니다. employees 테이블에 대한 DELETE 및 INSERT 권한과 HR\_CLERK 룰은 HR\_MGR 룰에 부여됩니다. 관리자는 HR\_MGR 룰을 부여 받고 이제 employees 테이블을 선택, 삭제, 삽입 및 갱신할 수 있습니다.

## 미리 정의된 룰

룰	포함된 권한
CONNECT	CREATE SESSION
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
DBA	대부분의 시스템 권한 및 기타 여러 가지 룰. 비관리자에게는 부여 안함.
SELECT_CATALOG_ROLE	시스템 권한은 없지만 HS_ADMIN_ROLE 및 데이터 딕셔너리에 대한 1,700개 이상의 객체 권한이 있음

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 미리 정의된 룰

데이터베이스 생성 스크립트를 실행할 때 오라클 데이터베이스에 대해 여러 가지 룰이 자동으로 정의됩니다. CONNECT는 Enterprise Manager를 사용하여 생성된 유저에게 자동으로 부여됩니다. 보안상의 이유로 오라클 데이터베이스 10.2.0 이후 버전에는 CONNECT 룰에 CREATE SESSION 권한만이 포함됩니다.

**주:** RESOURCE 룰 부여 시 UNLIMITED TABLESPACE 권한 부여가 포함됩니다.

### 기능 룰

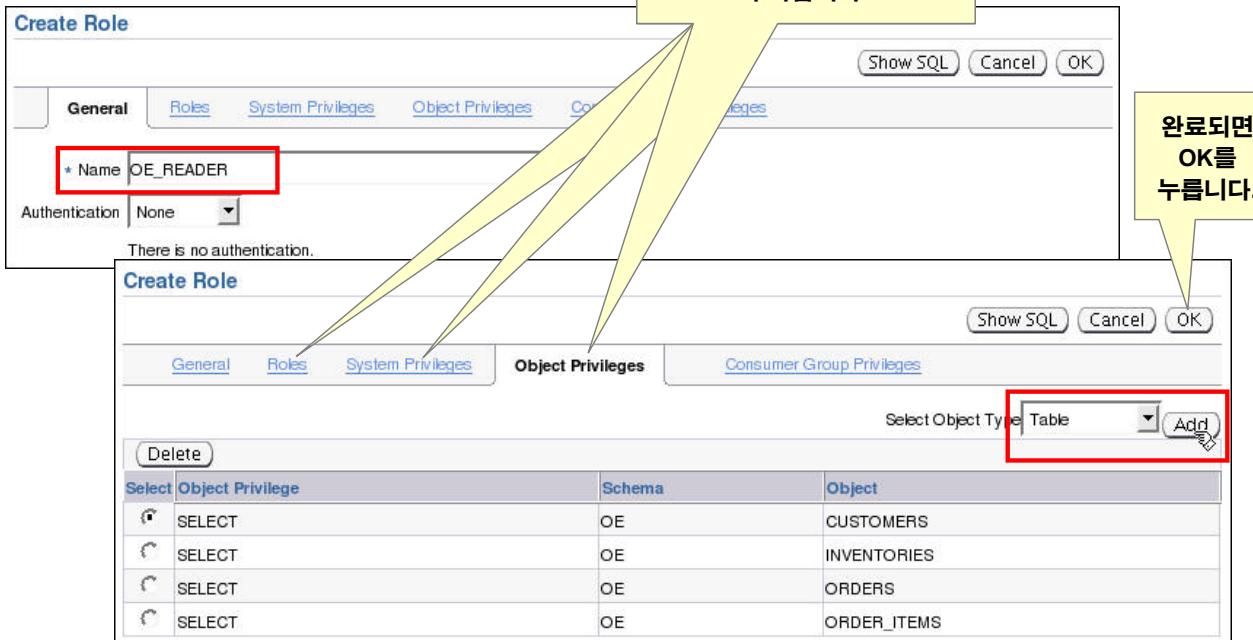
특수 기능 관리 권한을 부여하는 기타 룰은 해당 기능 설치 시 생성됩니다. 예를 들어, XDBADMIN에는 해당 기능이 설치된 경우 XML(Extensible Markup Language) 데이터베이스 관리에 필요한 권한이 포함되어 있습니다. AQ\_ADMINISTRATOR\_ROLE은 Advanced Queuing에 필요한 권한을 제공합니다. HS\_ADMIN\_ROLE에는 이기종 서비스 관리에 필요한 권한이 포함되어 있습니다.

실수로 필요한 기능을 비활성화할 수 있기 때문에 이러한 기능 룰에 부여된 권한을 Oracle 지원 센터의 도움 없이 변경해서는 안됩니다.

# 롤 생성

Server > Roles를 선택합니다.

적합한 탭에서 권한 및 룰을 추가합니다.



완료되면  
OK를  
누릅니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 롤 생성

룰(role)은 유저나 다른 룰에 부여된 관련 권한의 명명된 그룹입니다. DBA는 룰을 통해 권한을 관리합니다.

룰을 생성하려면 다음 단계를 수행하십시오.

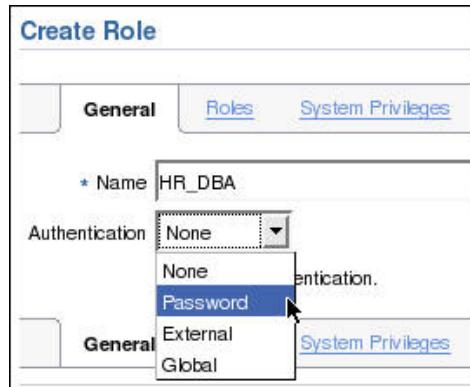
1. Enterprise Manager Database Control에서 Server 탭을 누르고 Security 머리글 아래에서 Roles를 누릅니다.
2. Create 버튼을 누릅니다.
3. General 탭에서 룰 이름을 입력합니다.
4. 선택적으로 시스템 권한, 객체 권한 및 기타 룰을 필요에 따라 추가합니다. 필요한 경우 나중에 룰을 편집하여 이러한 설정을 수정할 수 있습니다.
5. 완료되면 OK를 누릅니다.

## 보안 룰

- 룰은 기본값 이외의 값으로 설정하고 필요에 따라 활성화할 수 있습니다.

```
SET ROLE vacationdba;
```

- 룰은 인증을 통해 보호할 수 있습니다.



- 룰은 프로그래밍 방식으로도 보안을 유지할 수 있습니다.

```
CREATE ROLE secure_application_role  
IDENTIFIED USING <security_procedure_name>;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 보안 룰

대개 룰은 기본적으로 활성화되어 있습니다. 즉, 룰을 유저에게 할당하면 유저는 룰에 부여된 권한을 행사할 수 있습니다. 기본 룰은 연결 시에 유저에게 할당됩니다.

룰에 대해 다음을 수행할 수 있습니다.

- 룰을 비기본값으로 생성: 룰을 유저에게 부여할 때 DEFAULT 체크 박스의 선택을 해제합니다. 이제 유저가 룰의 권한을 행사하려면 룰을 명시적으로 활성화해야 합니다.
- 추가 인증이 필요한 룰 생성: 룰에 대한 기본 인증은 None이지만 이를 설정하기 전에 추가 인증이 필요하도록 설정할 수 있습니다.
- PL/SQL 프로시저를 성공적으로 실행해야만 활성화될 수 있는 보안 응용 프로그램 룰 생성: PL/SQL 프로시저에서는 유저의 네트워크 주소, 유저가 실행하고 있는 프로그램, 시간 또는 사용 권한 그룹의 보안을 제대로 유지하는 데 필요한 기타 요소를 검사할 수 있습니다.
- Oracle Database Vault 옵션을 사용하여 손쉽게 룰 관리: 보안 응용 프로그램 룰이 간소화되며 기존의 룰을 보다 세부적으로 제한할 수 있습니다.

# 유저에게 룰 할당

Modify Roles

Available Roles	Selected Roles
JAVA_DEPLOY JMXSERVER LOGSTDBY_ADMINISTRATOR MGMT_USER OEM_Advisor OEM_Monitor <b>OE_READER</b> OLAP_TRACE_USER OLAP_DBA OLAP_USER	CONNECT

Available Roles: JAVA\_DEPLOY, JMXSERVER, LOGSTDBY\_ADMINISTRATOR, MGMT\_USER, OEM\_Advisor, OEM\_Monitor, **OE\_READER**, OLAP\_TRACE\_USER, OLAP\_DBA, OLAP\_USER

Selected Roles: CONNECT

Buttons: Cancel, OK, Move, Move All, Remove, Remove All



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 유저에게 룰 할당

룰을 사용하여 데이터베이스 권한을 관리할 수 있습니다. 룰에 권한을 추가한 다음 이 룰을 유저에게 부여할 수 있습니다. 그러면 유저가 룰을 활성화하여 룰이 부여한 권한을 행사할 수 있습니다. 룰에는 이 룰에 부여된 모든 권한과 이 룰에 부여된 다른 룰의 모든 권한이 포함되어 있습니다.

Enterprise Manager에서는 기본적으로 새 유저에게 CONNECT 룰을 자동으로 부여합니다. 이를 통해 유저가 데이터베이스에 연결할 수 있고 유저가 소유한 스키마에 데이터베이스 객체를 생성할 수 있습니다.

유저에게 룰을 할당하려면 다음을 수행하십시오.

1. Enterprise Manager Database Control에서 Server 탭을 누르고 Security 머리글 아래에서 Users를 누릅니다.
2. 유저를 선택하고 Edit 버튼을 누릅니다.
3. Roles 탭을 누른 다음 Edit List 버튼을 누릅니다.
4. Available Roles에서 원하는 룰을 선택하고 Selected Roles로 이동합니다.
5. 적합한 룰을 모두 할당한 후에 OK 버튼을 누릅니다.

## 퀴즈

Oracle Database 11g에서 생성한 모든 암호는 기본적으로  
대소문자를 구분하지 않습니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 퀴즈

데이터베이스 툴의 특성은 무엇입니까?

1. 활성화하거나 비활성화할 수 있음
2. 시스템 및 객체 권한으로 구성할 수 있음
3. 생성자에 의해 소유됨
4. 암호로 보호할 수 없음

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1, 2

# 프로파일 및 유저

**유저에게는 한 번에 하나의  
프로파일만 할당됩니다.**

## 프로파일:

- 리소스 소비 제어
- 계정 상태 및 암호  
만료 관리

Create Profile

Show SQL Cancel OK

General Password

\* Name LIMITED\_USER

**Details**

CPU/Session (Sec./100)	1000	▼
CPU/Call (Sec./100)	UNLIMITED	▼
Connect Time (Minutes)	DEFAULT	▼
Idle Time (Minutes)	60	▼

**Database Services**

Concurrent Sessions (Per User)	DEFAULT	▼
Reads/Session (Blocks)	DEFAULT	▼
Reads/Call (Blocks)	DEFAULT	▼
Private SGA (KBytes)	DEFAULT	▼
Composite Limit (Service Units)	DEFAULT	▼

**주: 프로파일이 리소스 제한을 적용하려면 RESOURCE\_LIMIT를 TRUE로 설정해야 합니다.**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 프로파일 및 유저

프로파일은 데이터베이스 사용 및 instance 리소스에 명명된 리소스 제한 집합을 적용합니다. 프로파일은 또한 계정 상태를 관리하고 유저의 암호에 길이, 만료 시간 등의 제한을 지정합니다. 모든 유저에게 프로파일이 할당되며 언제나 하나의 프로파일에만 속할 수 있습니다. 유저 프로파일을 변경할 때 유저가 이미 로그인되어 있는 경우에는 변경 내용이 다음 로그인부터 적용됩니다.

DEFAULT 프로파일은 다른 모든 프로파일에 대한 기본입니다. 슬라이드에서 볼 수 있듯이 프로파일에 대한 제한은 CPU/Session에 지정된 것과 같이 암시적으로 지정하거나 CPU/Call에 지정된 것과 같이 무제한으로 지정할 수도 있고 Connect Time에 지정된 것과 같이 DEFAULT 프로파일에 있는 설정을 참조하도록 지정할 수 있습니다.

프로파일은 RESOURCE\_LIMIT 초기화 파라미터가 TRUE로 설정되어 있지 않으면 유저에게 리소스 제한을 적용할 수 없습니다. RESOURCE\_LIMIT가 기본값인 FALSE로 설정되어 있는 경우 프로파일 리소스 제한은 무시됩니다. 프로파일 암호 설정은 항상 적용됩니다.

프로파일을 통해 관리자는 다음과 같은 시스템 리소스를 제어할 수 있습니다.

- **CPU:** CPU 리소스를 세션 또는 호출 단위로 제한할 수 있습니다. CPU/Session을 1,000으로 제한하면 해당 프로파일을 사용하는 개별 세션이 CPU 시간을 10초 이상 소모하는 경우 (CPU 시간 제한은 1/100초 단위임) 해당 세션은 오류 메시지를 수신한 후 로그오프됩니다.  
ORA-02392: exceeded session limit on CPU usage, you are being logged off

## 프로파일 및 유저(계속)

호출 단위로 제한해도 결과는 같지만 유저의 전체 세션을 제한하는 대신 단일 명령이 너무 많은 CPU를 소모하지 않도록 합니다. CPU/Call이 제한되어 있을 때 유저가 제한을 초과하면 명령이 중지됩니다. 그리고 유저가 다음과 같은 오류 메시지를 받게 됩니다.

ORA-02393: exceeded call limit on CPU usage

- **네트워크/메모리:** 각 데이터베이스 세션은 시스템 메모리 리소스 및 네트워크 리소스(세션이 서버에 대한 로컬이 아닌 유저에 해당하는 경우)를 소모합니다. 다음을 지정할 수 있습니다.
  - **Connect Time:** 유저에게 허용되는 연결 시간(분)을 나타냅니다. 이 시간을 초과하면 자동으로 로그오프됩니다.
  - **Idle Time:** 유저의 세션이 Idle 상태를 유지할 수 있는 시간(분)을 나타냅니다. 이 시간을 초과하면 자동으로 로그오프됩니다. Idle 시간은 서버 프로세스에 대해서만 계산되며 응용 프로그램 작업은 고려하지 않습니다. IDLE\_TIME 제한은 장기적으로 실행하는 query 및 다른 작업의 영향을 받지 않습니다.
  - **Concurrent Sessions:** 데이터베이스 유저 계정을 사용하여 생성할 수 있는 동시 세션 수를 나타냅니다.
  - **Private SGA:** SGA(System Global Area) 내에서 정렬, 비트맵 병합 등에 소모되는 공간의 양을 제한합니다. 이 제한은 세션에서 Shared Server를 사용하는 경우에만 적용됩니다. Shared Server는 "Oracle 네트워크 환경 구성" 단원에서 설명합니다.
- **디스크 I/O:** 유저가 세션 단위 레벨 또는 호출 단위 레벨에서 읽을 수 있는 데이터의 양을 제한합니다. Reads/Session 및 Reads/Call은 메모리와 디스크에서 전체 읽기 횟수를 제한합니다. 따라서 I/O 중심 명령문이 메모리와 디스크를 과도하게 사용하지 않도록 보장할 수 있습니다.

프로파일은 또한 조합 제한을 허용합니다. Composite Limit는 CPU/Session, Reads/Session, Connect Time 및 Private SGA의 가중치 조합을 기반으로 합니다. 조합 제한에 대해서는 *Oracle Database Security Guide*에 자세히 설명되어 있습니다.

프로파일을 생성하려면 Server 탭을 누른 다음 Security 머리를 아래에서 Profiles를 누릅니다. Profiles 페이지에서 Create 버튼을 누릅니다.

**주:** Resource Manager는 여러 프로파일 설정을 대체하는 하나의 대안입니다. Resource Manager에 대한 자세한 내용은 *Oracle Database Administrator's Guide*를 참조하십시오.

## 암호 보안 기능 구현



**주: SYS, SYSMAN 및 DBSNMP 암호를 만료시키고 계정을 Lock하는 프로파일은 사용하지 마십시오.**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 암호 보안 기능 구현

Oracle 암호 관리는 유저 프로파일과 함께 구현됩니다. 프로파일은 여러 표준 보안 기능을 제공할 수 있습니다.

**계정 Lock:** 유저가 지정한 시도 횟수 내에 시스템에 로그인하지 못하면 일정 기간 동안 계정을 자동으로 Lock할 수 있습니다.

- **FAILED\_LOGIN\_ATTEMPTS:** 계정이 Lock되기 전의 로그인 시도 실패 횟수를 지정합니다.
- **PASSWORD\_LOCK\_TIME:** 지정된 횟수의 로그인 시도가 실패하면 계정이 Lock된 상태로 유지되는 기간(일)을 지정합니다.

**암호 aging 및 만료:** 유저 암호에 수명이 있어서 암호가 만료되면 암호를 변경해야 합니다.

- **PASSWORD\_LIFE\_TIME:** 암호가 만료되기 전에 사용할 수 있는 실행 시간(일)입니다.
- **PASSWORD\_GRACE\_TIME:** 암호 만료 후 처음 로그인할 때부터 시작되는 암호 변경을 위한 유예 기간(일)을 지정합니다.

**주:** 암호를 만료시키고 SYS, SYSMAN 및 DBSNMP 계정을 Lock하면 Enterprise Manager가 올바르게 작동하지 않습니다. 응용 프로그램은 "암호 만료" 경고 메시지를 감지하고 암호 변경을 처리해야 합니다. 그렇지 않으면 유저가 이유를 알지 못한 채 유예 기간이 만료되고 lock되게 됩니다.

## 암호 보안 기능 구현(계속)

**과거 암호 기록:** 지정한 기간 또는 암호 변경 횟수 동안 암호를 재사용하지 않았는지 확인하기 위해 새 암호를 검사합니다. 이 검사는 다음 파라미터 중 하나를 사용하여 구현할 수 있습니다.

- **PASSWORD\_REUSE\_TIME:** 주어진 일 수 동안 암호를 재사용할 수 없도록 지정합니다.
- **PASSWORD\_REUSE\_MAX:** 현재 암호를 재사용할 수 있기 전에 필요한 암호 변경 횟수를 지정합니다.

프로파일 파라미터의 값은 DEFAULT 프로파일에서 설정되거나 상속됩니다.

두 과거 암호 기록 파라미터의 값이 모두 UNLIMITED이면 오라클 데이터베이스는 두 값을 모두 무시합니다. 그러면 유저가 언제든지 암호를 재사용할 수 있는데 이는 보안상 좋지 않습니다.

두 파라미터가 모두 설정되어 있는 경우에는 두 조건을 모두 충족시켜야 암호를 재사용할 수 있습니다. 즉, 유저가 지정된 횟수만큼 암호를 변경했어야 하며 이전 암호를 마지막으로 사용한 이후로 지정된 기간(일)이 경과되었어야 합니다.

예를 들어, ALFRED 유저의 프로파일에서 PASSWORD\_REUSE\_MAX가 10으로, PASSWORD\_REUSE\_TIME이 30으로 설정되어 있는 경우 ALFRED는 암호를 10회 재설정하고 암호를 마지막으로 사용한 지 30일이 경과되지 않으면 암호를 재사용할 수 없습니다.

두 파라미터 중 하나는 숫자로 설정되어 있고 다른 하나는 UNLIMITED로 설정되어 있으면 유저는 암호를 재사용할 수 없습니다.

**암호 복잡성 확인:** 암호의 복잡성을 검사하여 특정 규칙을 만족하는지 확인합니다. 침입자가 암호를 추측하여 시스템에 침입할 수 없을 만큼 암호가 복잡한지 확인해야 합니다.

PASSWORD\_VERIFY\_FUNCTION 파라미터는 암호가 할당되기 전에 암호 복잡성 검사를 수행하는 PL/SQL 함수의 이름을 지정합니다. 암호 확인 함수는 SYS 유저가 소유해야 하며 Boolean 값(TRUE 또는 FALSE)을 반환해야 합니다. 다음 디렉토리에 있는 ut1pwdmg.sql 스크립트에서 모델 암호 확인 함수가 제공됩니다.

- Unix 및 Linux 플랫폼: \$ORACLE\_HOME/rdbms/admin
- Windows 플랫폼: %ORACLE\_HOME%\rdbms\admin

## 암호 프로파일 생성

**Create Profile**

**General** **Password** **Show SQL** **Cancel** **OK**

**Password**

Expire in (days)

Lock (days past expiration)

**History**

Number of passwords to keep

Number of days to keep for

**Complexity**

Complexity function

**Failed Login**

Number of failed login attempts to lock after

Number of days to lock for

Copyright © 2010, Oracle. All rights reserved.

### 암호 프로파일 생성

암호 프로파일을 생성하려면 Server 탭을 누른 다음 Security 머리글 아래에서 Profiles를 누릅니다. Profiles 페이지에서 Create 버튼을 누릅니다. Password 탭을 눌러 암호 제한을 설정합니다.

손전등 아이콘을 누르면 나오는 값 리스트에서 각 설정의 일반적인 값을 선택하거나 커스텀 값을 입력할 수 있습니다.

모든 시간대는 일 단위로 표시되지만 분수를 사용할 수도 있습니다. 예를 들어, 하루는 1,440분이므로 5/1,440는 5분에 해당합니다.

Enterprise Manager를 사용하여 기존 암호 프로파일을 편집할 수도 있습니다.

`utlpwdmg.sql` 스크립트가 실행된 경우 `VERIFY_FUNCTION` 및 `VERIFY_FUNCTION_11G` 함수를 사용할 수 있습니다. 자체 복합성 함수를 생성한 경우에는 해당 함수의 이름을 입력하면 됩니다. 함수 이름은 Select list에 표시되지 않습니다. 함수에서 런타임 오류가 발생하는 경우에는 유저가 암호를 변경할 수 없습니다.

### 암호 프로파일 삭제

Enterprise Manager에서 유저에 의해 사용되는 프로파일을 삭제할 수 없습니다. 그러나 예를 들어, SQL\*Plus에서 CASCADE 옵션을 사용하여 프로파일을 삭제하면 해당 프로파일을 가진 모든 유저가 자동으로 DEFAULT 프로파일에 할당됩니다.

## 제공된 암호 확인 함수: **VERIFY\_FUNCTION\_11G**

**VERIFY\_FUNCTION\_11G** 함수는 암호에 대해 다음을 확인합니다.

- 8자 이상임
- Username, Username에 숫자를 붙인 형태, Username을 거꾸로 한 형태가 아님
- 데이터베이스 이름 또는 데이터베이스 이름에 숫자를 붙인 형태가 아님
- 최소한 하나의 영문자와 하나의 숫자를 포함하는 문자열임
- 이전 암호와 최소한 3자 이상이 다름

참고: 이 함수를 템플리트로 사용하여 직접  
커스터마이즈된 암호 확인 함수를 생성할 수 있습니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 제공된 암호 확인 함수: **VERIFY\_FUNCTION\_11G**

Oracle 서버에서는 두 가지 암호 복합성 확인 함수(VERIFY\_FUNCTION 및 VERIFY\_FUNCTION\_11g)를 제공합니다. 이러한 함수는 <oracle\_home>/rdbms/admin/utlpwdmg.sql 스크립트를 사용하여 생성됩니다. VERIFY\_FUNCTION은 이전 버전에서 제공되는 암호 함수를 선호하는 유저를 위해 제공됩니다. 암호 복잡성 확인 함수는 SYS 스키마에서 생성되어야 합니다. 커스터마이즈된 암호 확인에 대한 템플리트로 사용될 수 있습니다.

VERIFY\_FUNCTION을 생성하는 것 외에도 utlpwdmg 스크립트는 DEFAULT 프로파일을 다음과 같은 ALTER PROFILE 명령으로 변경합니다.

```
ALTER PROFILE default LIMIT
  PASSWORD_LIFE_TIME 180
  PASSWORD_GRACE_TIME 7
  PASSWORD_REUSE_TIME UNLIMITED
  PASSWORD_REUSE_MAX UNLIMITED
  FAILED_LOGIN_ATTEMPTS 10
  PASSWORD_LOCK_TIME 1
  PASSWORD_VERIFY_FUNCTION verify_function_11g;
```

다른 프로파일을 지정하지 않은 경우 유저 생성 시 DEFAULT 프로파일이 할당됩니다.

## 유저에게 할당량 할당

**UNLIMITED TABLESPACE** 시스템 권한이 없는 유저가 테이블스페이스에 객체를 생성하려면 할당량이 제공되어야 합니다.  
할당량에는 다음 옵션이 있습니다.

- 특정 값(MB 또는 KB)
- Unlimited

Tablespace	Quota	Value	Unit
EXAMPLE	Value	20	MBytes
INVENTORY	None	0	MBytes
SYSAUX	None	0	MBytes
SYSTEM	None	0	MBytes
TEMP	None	0	MBytes
UNDOTBS1	None	0	MBytes
USERS (Default)	Unlimited	0	MBytes

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 유저에게 할당량 할당

할당량(Quota)은 주어진 테이블스페이스에 허용된 공간입니다. 기본적으로 유저의 테이블스페이스에는 할당량이 없습니다. 테이블스페이스에서 유저에게 할당량을 제공하는 옵션에는 세 가지가 있습니다.

- **Unlimited:** 이 옵션을 사용하면 유저가 테이블스페이스에서 사용할 수 있는 만큼 공간을 사용할 수 있습니다.
- **Value:** 유저가 사용할 수 있는 값(KB 또는 MB)입니다. 지정한 값이 현재 테이블스페이스에서 사용할 수 있는 공간보다 크거나 작을 수 있기 때문에 이 옵션을 사용해도 해당 유저에게 지정한 만큼의 공간이 확보되는 것은 아닙니다.
- **UNLIMITED TABLESPACE** 시스템 권한: 이 시스템 권한을 사용하면 모든 개별 테이블스페이스 할당량이 재정의되고 유저에게 SYSTEM 및 SYSAUX를 포함한 모든 테이블스페이스에 대해 무제한 할당량이 주어집니다. 이 권한을 부여할 때는 주의해야 합니다.

주: RESOURCE 룰을 부여하면 이 권한도 부여됩니다.

## 유저에게 할당량 할당(계속)

SYSTEM 또는 SYSAUX 테이블스페이스의 유저에게는 할당량을 제공해서는 안됩니다.

일반적으로 SYS 및 SYSTEM 유저만 SYSTEM 또는 SYSAUX 테이블스페이스에 객체를 생성할 수 있어야 합니다.

유저가 할당 받은 임시 테이블스페이스나 기타 언두 테이블스페이스에는 할당량이 필요 없습니다. 오라클 데이터베이스에서 데이터를 삽입, 생성 및 삭제할 때는 할당량이 필요하지 않습니다. 할당량이 필요한 유저는 데이터베이스 객체를 소유하는 계정뿐입니다. 일반적으로는 응용 프로그램 코드를 설치할 때 설치자가 객체를 소유하는 계정을 생성합니다. 이러한 계정에서만 할당량이 필요합니다. 다른 데이터베이스 유저에게는 할당량 없이 이러한 객체를 사용할 수 있는 권한을 부여할 수 있습니다.

- Oracle Instance는 유저가 세그먼트를 생성하거나 확장하면 할당량을 검사합니다.
- 유저 스키마에 할당되는 작업의 경우 테이블스페이스의 공간을 사용하는 작업만이 할당량 계산에 포함됩니다. 뷰 생성 또는 임시 테이블스페이스 사용 등과 같이 할당된 테이블스페이스의 공간을 사용하지 않는 작업은 할당량에 영향을 주지 않습니다.
- 할당량은 PURGE 절을 사용하여 유저가 소유한 객체를 삭제하거나 유저가 소유한 Recycle bin의 객체가 삭제될 때 다시 채워집니다.

# 최소 권한의 원칙 적용

- 데이터 딕셔너리 보호:

**07\_DICTIONARY\_ACCESSIBILITY=FALSE**

- PUBLIC에서 불필요한 권한 취소
- ACL(access control lists)을 사용하여 네트워크 액세스 제어
- 유저가 액세스할 수 있는 디렉토리 제한
- 관리 권한을 갖는 유저 제한
- 원격 데이터베이스 인증 제한:

**REMOTE\_OS\_AUTHENT=FALSE**

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 최소 권한의 원칙 적용

최소 권한의 원칙이란 유저가 작업을 효율적으로 완료하는 데 필요한 권한만 유저에게 부여한다는 의미입니다. 따라서 유저가 수정하거나 볼 수 있는 권한을 가져서는 안 되는 데이터를 실수로 또는 악의적으로 수정하거나 볼 수 있는 가능성이 줄어듭니다.

**데이터 딕셔너리 보호:** 07\_DICTIONARY\_ACCESSIBILITY 파라미터는 기본적으로 FALSE로 설정됩니다. 이 파라미터는 ANY TABLE 시스템 권한을 가진 유저가 데이터 딕셔너리 기본 테이블에 액세스하는 것을 방지하기 때문에 합당한 이유 없이 이 파라미터를 변경해서는 안됩니다. 또한 이 파라미터를 사용하면 SYS 유저가 SYSDBA로만 로그인할 수 있습니다.

**PUBLIC에서 불필요한 권한 취소:** 다양한 패키지는 이를 필요로 하는 응용 프로그램에서는 매우 유용하지만 안전하게 사용하려면 제대로 구성해야 합니다. PUBLIC에는 UTL\_SMTP, UTL\_TCP, UTL\_HTTP 및 UTL\_FILE 패키지에 대한 실행 권한이 부여됩니다.

Oracle Database 11g에서 네트워크 액세스는 특정 네트워크 서비스에 대한 특정 유저 액세스를 허용하도록 구성할 수 있는 ACL(Access Control List)에 의해 제어됩니다. 네트워크 액세스는 기본적으로 거부됩니다. 네트워크 액세스를 허용하려면 ACL을 생성해야 합니다. UTL\_FILE 액세스를 통한 파일은 두 가지 레벨, 즉 파일 및 디렉토리에 대한 권한이 있는 OS 레벨과 특정 파일 시스템 디렉토리에 대한 액세스를 허용하는 DIRECTORY 객체에 의해 데이터베이스 레벨에서 제어됩니다. 유저에게는 DIRECTORY 객체 읽기 또는 읽기/쓰기 권한이 부여될 수 있습니다. 다른 PL/SQL 패키지에 대한 실행 권한은 주의하여 제어해야 합니다.

## 최소 권한의 원칙 적용(계속)

잠재적으로 잘못 사용될 가능성이 있는 더 강력한 패키지는 다음과 같습니다.

- **UTL\_SMTP:** 데이터베이스를 SMTP(Simple Mail Transfer Protocol) 메일 서버로 사용하여 임의로 전자 메일 메시지를 보낼 수 있습니다. ACL을 사용하여 각 유저가 액세스할 수 있는 컴퓨터를 제어할 수 있습니다.
- **UTL\_TCP:** 데이터베이스 서버가 수신 또는 대기 중인 네트워크 서비스에 송신 네트워크 연결을 설정하도록 허용합니다. 그러면 데이터베이스 서버와 대기 중인 네트워크 서비스 간에 임의의 데이터를 전송할 수 있습니다. ACL을 사용하여 액세스를 제어할 수 있습니다.
- **UTL\_HTTP:** 데이터베이스 서버가 HTTP를 통해 데이터를 요청하고 검색하도록 허용합니다. 유저에게 이 패키지를 부여하면 데이터가 HTML의 형태로 악의적인 웹 사이트에 전송될 수 있습니다. ACL을 사용하여 액세스를 제한할 수 있습니다.
- **UTL\_FILE:** 잘못 구성하면 호스트 운영 체제의 모든 파일에 대해 텍스트 레벨의 액세스가 허용됩니다. 올바르게 구성하는 경우 이 패키지는 특정 디렉토리 위치에 대한 유저 액세스를 제한합니다.

**OS 디렉토리에 대한 액세스 제한:** DBA는 데이터베이스 내의 DIRECTORY 객체를 사용하여 디렉토리를 OS 경로에 매핑하고 개별 유저에게 해당 디렉토리에 대한 권한을 부여할 수 있습니다.

**관리 권한이 있는 유저 제한:** 데이터베이스 유저에게 필요 이상으로 많은 권한을 제공하지 마십시오. 비관리자에게는 DBA 룰을 부여해서는 안됩니다. 최소 권한을 부여하려면 다음 유형의 권한을 제한하십시오.

- 시스템 및 객체 권한 부여
- 데이터베이스에 대해 SYSDBA 및 SYSOPER와 같은 SYS 권한이 부여된 연결
- DROP ANY TABLE

과 같은 기타 DBA 유형의 권한원격 데이터베이스 인증 제한: REMOTE\_OS\_AUTHENT 파라미터는 기본적으로 FALSE로 설정됩니다. 모든 클라이언트가 유저를 올바르게 인증할 수 있도록 신뢰될 수 있는 경우를 제외하고 이 파라미터를 변경해서는 안됩니다.

Oracle Database 10g Release 2에서 제공되는 Secure External Password Store가 도입되면서 원격 OS 인증을 사용할 필요성이 거의 없어졌습니다.

원격 인증 프로세스에서는 다음과 같은 작업이 가능합니다.

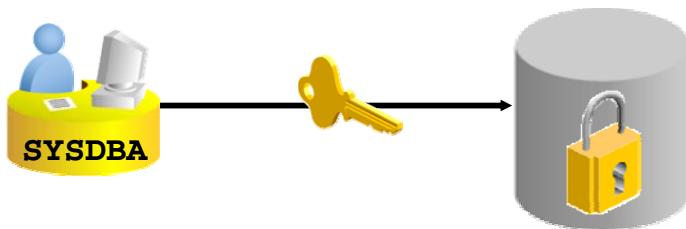
- 외부 인증으로 데이터베이스 유저가 인증됩니다.
- 원격 시스템에서 유저를 인증합니다.
- 유저가 추가 인증 없이 데이터베이스에 로그인합니다.

**주:** 권한을 취소한 경우 항상 응용 프로그램을 철저하게 테스트하십시오.

## 권한이 부여된 계정 보호

권한이 부여된 계정은 다음과 같은 방법으로 보호할 수 있습니다.

- 대소문자 구분 암호로 Password file을 사용합니다.
- 관리자 롤에 대해 강력한 인증을 활성화합니다.



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 데이터베이스 관리자 인증 설정

SYSDBA, SYSOPER 또는 SYSASM 권한이 있는 유저는 항상 인증을 받아야 합니다.

로컬로 연결할 때 이러한 유저는 권한이 있는 OS 그룹 멤버인 경우 로컬 OS에 의해 인증됩니다. 원격으로 연결하는 경우에는 Password file을 사용하여 권한이 있는 유저를 인증합니다. Password file이 구성되어 있으면 먼저 검사됩니다. Oracle Database 11g에서 이러한 암호는 대소문자를 구분합니다. Oracle Database 11g에는 원격 관리자 인증을 보다 안전하게 만들고 이러한 권한을 가진 유저들을 중앙 집중식으로 관리할 수 있도록 하는 여러 가지 방법을 제공합니다.

Database Configuration Assistant를 사용하여 데이터베이스를 생성하는 경우 Password file은 대소문자를 구분합니다. 이전 데이터베이스 버전에서 업그레이드하는 경우에는 원격 연결에 대해 Password file이 대소문자를 구분하도록 하십시오.

```
orapwd file=orapworcl entries=5 ignorecase=N
```

Password file이 취약하거나 여러 Password file을 유지 관리하는 것이 부담스러울 것으로 생각되는 경우에는 강력한 인증을 구현할 수 있습니다. 강력한 인증 방법을 사용하려는 경우에는 Advanced Security Option<sup>o</sup> 필요합니다. 강력한 인증에 대한 자세한 사항은 *Oracle Database Advanced Security Administrator's Guide*를 참조하십시오.

## 퀴즈

**최소 권한의 원칙만 적용하는 것은 오라클 데이터베이스를  
강화하는 데 충분하지 않습니다.**

- 1. 참**
- 2. 거짓**

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

**해답: 1**

## 퀴즈

**RESOURCE\_LIMIT가 기본값인 FALSE로 설정되어 있는 경우  
프로파일 암호 제한은 무시됩니다.**

- 1. 참**
- 2. 거짓**

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

**해답: 2**

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 데이터베이스 유저 계정 생성 및 관리:
  - 유저 인증
  - 기본 저장 영역(테이블스페이스) 할당
- 권한 부여 및 취소
- 룰 생성 및 관리
- 프로파일 생성 및 관리:
  - 표준 암호 보안 기능 구현
  - 유저별 리소스 사용량 제어

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 8 개요: 유저 관리

이 연습에서는 다음 내용을 다룹니다.

- 리소스 소비를 제한하는 프로파일 생성
- 다음과 같은 두 가지 를 생성:
  - HRCLERK
  - HRMANAGER
- 네 명의 새 유저 생성:
  - 한 명의 관리자 및 두 명의 직원
  - 후속 연습 세션을 위한 한 명의 스키마 유저

ORACLE®

Copyright © 2010, Oracle. All rights reserved.



# 데이터 동시성 관리

9

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

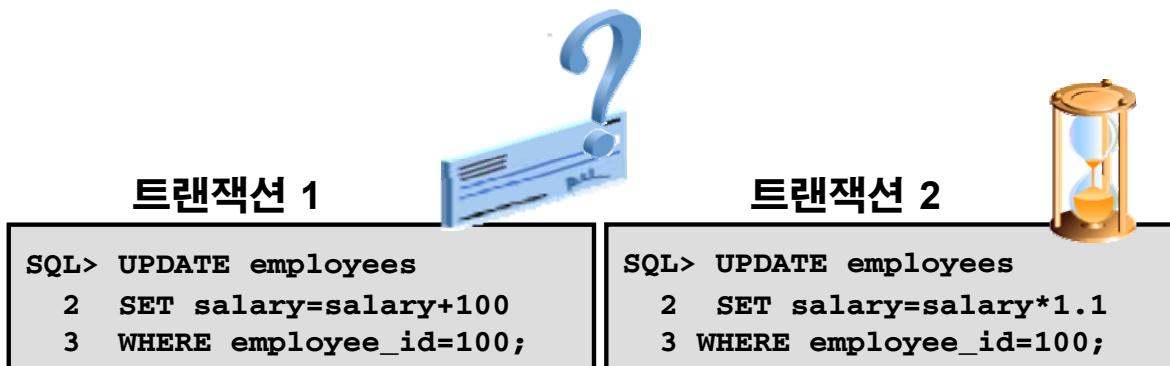
- Lock 메커니즘과 Oracle의 데이터 동시성 관리 방법 설명
- Lock 충돌 모니터 및 해결

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# Lock

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Lock

데이터베이스에서 세션이 데이터를 수정할 수 있도록 하려면 먼저 해당 세션이 수정하고 있는 데이터를 lock해야 합니다. Lock이 해제될 때까지 Lock된 데이터를 다른 트랜잭션에서 수정할 수 없도록 Lock은 세션에 배타적 데이터 제어를 제공합니다.

트랜잭션을 통해 개별 데이터 행, 다중 행 또는 전체 테이블까지 Lock할 수 있습니다. 오라클 데이터베이스에서는 수동 및 자동 Lock을 모두 지원합니다. 자동으로 획득한 Lock은 다른 트랜잭션과의 잠재적 충돌을 최소화하기 위해 항상 가장 낮은 Lock 레벨을 선택합니다.

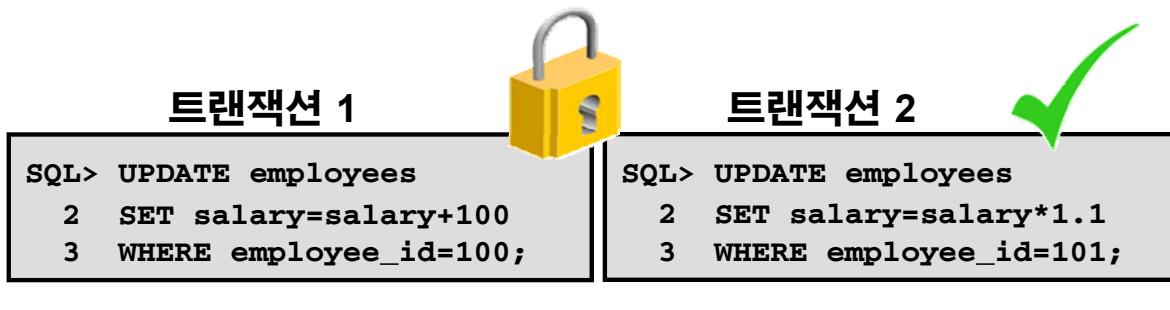
**주:** 내부 일관성을 유지 관리하기 위해 Oracle Instance에서는 다양한 유형의 Lock이 사용됩니다. 이 과정에서는 행과 테이블을 보호하기 위해 사용되는 Lock에 대해서만 중점적으로 다릅니다.

# Lock 메커니즘

- 높은 레벨의 데이터 동시성:
  - 삽입, 갱신 및 삭제에 대한 행 레벨 Lock
  - query에 lock이 필요 없음
- 자동 큐 관리
- 트랜잭션이 종료될 때까지 lock 보유(COMMIT 또는 ROLLBACK 작업 사용)

## 예제

employee\_id 100 및 101에 대한 행이 동일한 블록에 상주한다고 가정합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Lock 메커니즘

Lock 메커니즘의 목적은 데이터베이스 내에서 최대한의 데이터 동시성을 제공하는 것입니다. 데이터를 수정하는 트랜잭션은 블록 레벨 또는 테이블 레벨 lock이 아닌 행 레벨 lock을 획득합니다. 객체에 대한 수정(예: 테이블 이동)은 전체 데이터베이스 또는 스키마 lock이 아닌 객체 레벨 lock을 획득합니다.

데이터 query에는 Lock이 필요하지 않으며 다른 유저가 데이터를 Lock한 경우에도 query가 성공합니다. Lock된 데이터에는 항상 언두 정보로부터 재생성한 원래의 Lock 이전 값이 표시됩니다.

다중 트랜잭션에서 같은 리소스를 Lock해야 하는 경우 첫번째로 Lock을 요청한 트랜잭션이 Lock을 획득합니다. 다른 트랜잭션은 첫번째 트랜잭션이 완료될 때까지 대기합니다.

큐 메커니즘은 자동으로 수행되므로 관리자의 상호 작용이 필요하지 않습니다.

트랜잭션이 완료되면(즉, COMMIT 또는 ROLLBACK이 실행될 때) 모든 Lock이 해제됩니다.

트랜잭션이 실패하면 실패한 트랜잭션의 변경 사항을 자동으로 롤백하는 동일한 백그라운드 프로세스에서 해당 트랜잭션이 보유한 lock을 모두 해제합니다.

## 데이터 동시성

시간: 09:00:00	트랜잭션 1	<code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=100;</code>
	트랜잭션 2	<code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=101;</code>
	트랜잭션 3	<code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=102;</code>
	...	...
	트랜잭션 x	<code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=xxx;</code>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 데이터 동시성

Lock 메커니즘의 기본값은 하위단계(fine-grain)의 행 레벨 Lock 모드입니다. 다른 트랜잭션은 서로 간섭하지 않고 같은 테이블 내에서 다른 행을 갱신할 수 있습니다.

기본 모델은 행 레벨에서 Lock하는 것이지만 오라클 데이터베이스는 필요한 경우 다음과 같이 더 높은 레벨에서의 수동 Lock을 지원합니다.

```
SQL> LOCK TABLE employees IN EXCLUSIVE MODE;
Table(s) Locked.
```

위 명령문을 사용하면 lock된 테이블에서 행을 갱신하려는 다른 트랜잭션은 lock 요청을 실행한 트랜잭션이 완료될 때까지 대기해야 합니다. EXCLUSIVE는 가장 엄격한 lock 모드입니다.

다음은 기타 lock 모드입니다.

- **ROW SHARE:** lock된 테이블에 대한 동시 액세스를 허용하지만 세션이 배타적 액세스를 위해 전체 테이블을 lock하는 것은 금지합니다.
- **ROW EXCLUSIVE:** ROW SHARE와 동일하지만, SHARE 모드에서도 lock하는 것을 금지합니다. ROW EXCLUSIVE lock은 데이터 갱신, 삽입 또는 삭제 시 자동으로 획득됩니다. ROW EXCLUSIVE lock을 사용하면 여러 번 읽고 한 번 쓸 수 있습니다.
- **SHARE:** 동시 query는 허용하지만 Lock된 테이블에 대한 갱신은 금지합니다. 테이블에서 인덱스를 생성하려면 SHARE Lock이 필요하며 자동으로 요청됩니다. 그러나 인덱스를 온라인에서 생성하려면 인덱스를 작성할 때 ROW SHARE Lock을 사용해야 합니다.

## 데이터 동시성(계속)

Share Lock을 사용하면 여러 번 읽을 수 있지만 쓸 수는 없습니다. SHARE Lock은 상위 테이블에 대한 Foreign Key 제약 조건이 지정된 하위 테이블이 있는 상위 테이블에서 행을 삭제하거나 갱신할 때 투명하게 사용되기도 합니다.

- **SHARE ROW EXCLUSIVE:** 전체 테이블을 query하는 데 사용되며 다른 유저가 테이블의 행을 query하는 것은 허용하지만 해당 테이블을 SHARE 모드에서 lock하거나 행을 갱신하는 것은 금지합니다.
- **EXCLUSIVE:** lock된 테이블에서의 query는 허용하지만 해당 테이블에서의 다른 작업은 금지합니다. 테이블을 삭제하려면 EXCLUSIVE lock이 필요합니다.

Lock에 대한 다른 요청과 마찬가지로 수동 Lock 문은 이미 Lock을 갖고 있거나 이전에 Lock을 요청한 모든 세션이 보유한 Lock을 해제할 때까지 대기합니다. LOCK 명령은 대기 동작을 제어하는 특수 인수인 NOWAIT를 받아들입니다.

NOWAIT는 지정한 테이블이 다른 세션에서 이미 lock된 경우 유저에게 바로 제어를 반환합니다.

```
SQL> LOCK TABLE hr.employees IN SHARE MODE NOWAIT;
LOCK TABLE hr.employees IN SHARE MODE NOWAIT
*
ERROR at line 1:
ORA-00054: resource busy and acquire with NOWAIT specified
```

대개 수동으로 객체를 lock 할 필요는 없습니다. 자동 lock 메커니즘은 대부분의 응용 프로그램에 필요한 데이터 동시성을 제공합니다. 오라클에서는 응용 프로그램을 개발하는 경우 등에 수동 Lock을 사용하지 말 것을 권장합니다. Lock 레벨이 불필요하게 높은 경우 심각한 성능 문제가 발생하게 됩니다.

# DML Lock

## 트랜잭션 1

```
SQL> UPDATE employees  
  2  SET salary=salary*1.1  
  3  WHERE employee_id= 107;  
1 row updated.
```

## 트랜잭션 2

```
SQL> UPDATE employees  
  2  SET salary=salary*1.1  
  3  WHERE employee_id= 106;  
1 row updated.
```

각 DML 트랜잭션은 다음 두 개의 locks을 획득해야 합니다.

- 갱신 중인 행에 대한 EXCLUSIVE 행 Lock
- ROW EXCLUSIVE (RX) 모드에서 행을 포함한 테이블에 대한 테이블 lock (TM)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## DML Lock

각 DML 트랜잭션은 다음 두 개의 Lock을 획득합니다.

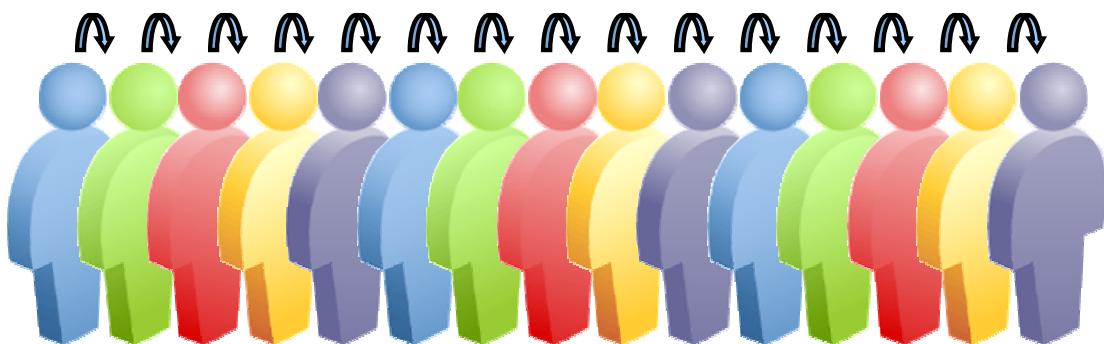
- 갱신 중인 행에 대한 EXCLUSIVE 행 Lock
- ROW EXCLUSIVE (RX) 모드에서 갱신 중인 테이블에 대한 테이블 lock (TM)입니다.  
이것은 변경 작업이 수행될 때 다른 세션에서 전체 테이블을 삭제하거나 truncate하기 위해 Lock하는 것을 방지합니다. 이 모드는 하위 배타적 테이블 lock (SX)이라고도 합니다.

테이블에 대한 ROW EXCLUSIVE Lock은 DDL이 커밋되지 않은 트랜잭션 도중 딕셔너리 메타 데이터를 변경하지 못하도록 합니다. 이를 통해 트랜잭션의 전체 실행에서 딕셔너리 무결성 및 읽기 일관성이 유지됩니다.

# Enqueue 메커니즘

Enqueue 메커니즘을 통해 다음을 추적할 수 있습니다.

- lock을 대기하는 세션
- 요청된 Lock 모드
- 세션에서 Lock을 요청한 순서



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## Enqueue 메커니즘

lock 요청은 자동으로 큐에 저장됩니다. Lock을 보유하는 트랜잭션이 완료되면 대기 중인 다음 세션이 Lock을 보유하게 됩니다.

Enqueue 메커니즘은 lock이 요청된 순서와 요청된 lock 모드를 추적합니다.

Lock을 이미 보유한 세션은 큐의 끝으로 이동하지 않고 해당 Lock의 변환을 요청할 수 있습니다. 예를 들어, 세션이 테이블에 대한 SHARE lock을 보유하는 경우 SHARE을 EXCLUSIVE Lock으로 변환하도록 요청할 수 있습니다. 테이블에 EXCLUSIVE 또는 SHARE lock을 이미 가지고 있는 다른 트랜잭션이 없는 경우, SHARE lock을 보유한 세션은 다시 큐에서 대기하지 않고 EXCLUSIVE lock이 부여됩니다.

**주:** Enqueue 메커니즘의 대기자에는 두 가지 범주가 있는데, 하나는 공유 소유권 없이 대기하는 대기자이고 다른 하나는 Lock 레벨을 Escalate하지 않는 공유 소유권이 있는 대기자입니다.

두 번째 범주의 대기자를 **변환자(converter)**라고 하며, 대기한 시간이 적더라도 항상 일반 대기자보다 높은 우선 순위가 부여됩니다.

# Lock 충돌

## 트랜잭션 1

## 시간

## 트랜잭션 2

UPDATE employees SET salary=salary+100 WHERE employee_id=100; 1 row updated.	9:00:00	UPDATE employees SET salary=salary+100 WHERE employee_id=101; 1 row updated.
UPDATE employees SET COMMISION_PCT=2 WHERE employee_id=101; Lock 충돌로 세션이 enqueue된 상태로 대기합니다.	9:00:05	SELECT sum(salary) FROM employees; SUM(SALARY) ----- 692634
세션이 계속 대기 중입니다!	16:30:00	지난 7시간 30분 동안 선택, 삽입, 갱신 및 삭제 작업이 많았지만 커밋이나 롤백이 없었습니다.
1 row updated. 세션이 계속됩니다.	16:30:01	commit;

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Lock 충돌

Lock 충돌이 자주 발생하지만 대개 시간이 지나면서 Enqueue 메커니즘으로 해결됩니다. lock 충돌 시 관리자의 개입이 필요한 경우도 있습니다. 슬라이드의 예제에서는 트랜잭션 2가 9:00:00에 단일 행에 대한 lock을 획득하고 커밋하지 않아 lock이 그대로 유지되고 있습니다. 트랜잭션 1이 전체 테이블을 갱신하려고 9:00:05에 모든 행에 대한 Lock을 요청했는데 트랜잭션 2가 16:30:01에 커밋할 때까지 트랜잭션 2에 의해 트랜잭션 1이 차단되었습니다.

이러한 경우 트랜잭션 1을 수행하려는 유저가 관리자에게 문의하면 DBA가 충돌을 감지하여 해결해야 합니다.

# Lock 충돌 유발 원인

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



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## Lock 충돌 유발 원인

Lock 충돌의 가장 일반적인 원인은 커밋되지 않은 변경 사항이지만 다음과 같은 다른 몇 가지 유발 원인이 더 있습니다.

- **장기 실행 트랜잭션:** 많은 응용 프로그램에서는 일괄 처리를 사용하여 대량 행을 수행합니다. 이러한 일괄 작업은 대개 유저 작업이 없거나 적은 시간에 일정이 잡히지만 시간이 너무 오래 걸려서 작업이 단기간 내에 끝나지 않는 경우도 있습니다. Lock 충돌은 트랜잭션과 일괄 처리가 동시에 수행될 때 일반적으로 발생합니다.
- **필요 이상으로 높은 lock 레벨:** 일부 데이터베이스는 행 레벨 lock을 지원하지 않습니다. Oracle은 1988년에 Release 6에서 행 레벨 lock을 추가했습니다. 따라서 일부 데이터베이스는 여전히 페이지 또는 테이블 레벨에서 Lock됩니다. 다양한 데이터베이스에서 실행할 수 있는 응용 프로그램을 작성하는 개발자는 오라클 데이터베이스가 이러한 저기능 데이터베이스 시스템과 유사하게 작동하도록 인위적으로 높은 Lock 레벨을 사용하여 응용 프로그램을 작성하기도 합니다. 또한 Oracle을 처음 사용하는 개발자가 때때로 오라클 데이터베이스에서 필요한 것보다 더 높은 Lock 레벨로 불필요한 코딩 작업을 수행하는 경우가 있습니다.

# Lock 충돌 감지

Performance 페이지에서 Blocking Sessions를 선택합니다.

Blocking Sessions											Page Refreshed Aug 18, 2008 11:04:23 PM MDT			Refresh
		Sessions Blocked	Session ID	Serial Number	SQL ID	Wait Class	Wait Event	P1 Value	P2 Value	P3 Value	Seconds In Wait			
<a href="#">View Session</a>		<a href="#">Kill Session</a>												
<a href="#">Expand All</a>		<a href="#">Collapse All</a>												
Select Username	▼ Blocking Sessions													
BERNST	▼	1	<a href="#">114</a>	33091		Idle	<a href="#">SQL*Net message from client</a>	16508152321	0		89			
SMAVRIS	▼	0	<a href="#">124</a>	46897	<a href="#">Otqkttcvhr5fcf</a>	Application	<a href="#">eng: TX - row lock contention</a>	1415053318	65545	3085	69			

Session ID 링크를 눌러 실제 SQL 문을 포함하여 lock 세션에 대한 정보를 확인합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Lock 충돌 감지

Enterprise Manager의 Blocking Sessions 페이지를 사용하여 lock 충돌을 찾습니다. 충돌 Lock 요청은 Lock을 보유하는 세션이 맨 위에 있고 그 아래 Lock에 대해 enqueue된 세션이 있는 계층적 레이아웃으로 표시됩니다.

충돌과 관련된 각 세션에 대해 Username, 세션 ID 및 세션이 대기한 시간(초)을 표시합니다. 세션 ID로 드릴 다운하여 세션에서 현재 실행 중이거나 요청 중인 실제 SQL 문을 확인합니다.

ADDM(Automatic Database Diagnostic Monitor)도 자동으로 lock 충돌을 감지하고 비효율적인 lock 상황을 알려줄 수 있습니다.

# Lock 충돌 해결

Lock 충돌을 해결하려면 다음을 수행하십시오.

- lock을 보유하는 세션을 커밋 또는 롤백합니다.
- Lock을 보유하는 세션을 즉시 종료합니다(비상시 사용).



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Lock 충돌 해결

lock 충돌을 해결하려면 lock을 보유하는 세션에서 lock을 해제해야 합니다. 세션에서 lock을 해제하는 가장 좋은 방법은 유저에게 연락하여 해당 트랜잭션 완료를 요청하는 것입니다.

비상시 관리자가 Kill Session 버튼을 눌러 Lock을 보유하는 세션을 종료할 수 있습니다. 하지만 이 방법으로 세션을 종료하면 현재 트랜잭션 내의 모든 작업이 손실(롤백)됨을 유념하십시오. 세션이 종료된 유저는 다시 로그인하여 세션의 마지막 커밋이 종료된 이후의 모든 작업을 리두해야 합니다.

세션이 종료된 유저가 다음에 SQL 문을 실행하려고 하면 다음 오류 메시지가 표시됩니다.

ORA-03135: connection lost contact

주: PMON 세션 스나이퍼는 Idle 타임아웃으로 인해 세션을 자동으로 종료할 수 있으며, 이는 프로파일 또는 Resource Manager를 통해 수행될 수 있습니다.

# SQL을 사용하여 Lock 충돌 해결

SQL 문은 차단된 세션을 파악하여 종료하는 데 사용될 수 있습니다.

1

```
SQL> select SID, SERIAL#, USERNAME  
      from V$SESSION where SID in  
(select BLOCKING_SESSION from V$SESSION)
```

결과:

SID	SERIAL#	USERNAME
144	8982	HR

2

```
SQL> alter system kill session '144,8982' immediate;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## SQL 을 사용하여 Lock 충돌 해결

Enterprise Manager에서 수행되는 대부분의 다른 작업과 마찬가지로 SQL 문을 실행하여 세션 조작을 수행할 수 있습니다. V\$SESSION 테이블은 연결된 모든 세션에 대한 세부 정보를 포함합니다. BLOCKING\_SESSION의 값은 차단 중인 세션의 세션 ID입니다. SID 및 SERIAL#(여기서 SID는 차단하는 세션 ID와 일치)에 대해 query하는 경우 kill session 작업을 수행하는 데 필요한 정보를 얻게 됩니다.

주: Database Resource Manager를 사용하면 다른 세션을 차단하고 Idle 상태인 세션에서 자동으로 로그아웃할 수 있습니다.

# Deadlock

트랜잭션 1		트랜잭션 2
<pre>UPDATE employees SET salary = salary * 1.1 WHERE employee_id = 1000;</pre>	9:00	<pre>UPDATE employees SET manager = 1342 WHERE employee_id = 2000;</pre>
<pre>UPDATE employees SET salary = salary * 1.1 WHERE employee_id = 2000;</pre>	9:15	<pre>UPDATE employees SET manager = 1342 WHERE employee_id = 1000;</pre>
<pre>ORA-00060: Deadlock detected while waiting for resource</pre>	9:16	

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Deadlock

Deadlock은 Lock 충돌의 특별한 예입니다. Deadlock은 둘 이상의 세션이 각각 Lock된 데이터를 서로 대기하고 있을 때 발생합니다. 각 세션이 다른 세션을 대기하고 있기 때문에 어떤 세션도 트랜잭션을 완료하여 충돌을 해결할 수 없습니다.

오라클 데이터베이스는 Deadlock을 자동으로 감지하여 오류가 발생한 명령문을 종료합니다. 해당 오류에 대한 적절한 응답은 커밋 또는 롤백이며, 다른 세션에서 트랜잭션을 계속 진행할 수 있도록 해당 세션에서 다른 lock을 해제합니다.

슬라이드의 예제에서 트랜잭션 1은 오류가 감지된 Deadlock에 대한 응답으로 커밋 또는 롤백해야 합니다. 커밋한 경우에는 두번째 갱신을 다시 제출(submit)해야 트랜잭션을 완료할 수 있으며 롤백을 수행한 경우에는 두 명령문을 모두 다시 제출(submit)하여 트랜잭션을 완료해야 합니다.

## 퀴즈

Lock 메커니즘의 기본값은 하위단계(fine-grain)의 행 레벨 Lock 모드입니다.

1. 참
2. 거짓

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1

## 퀴즈

Deadlock 발생 시, 오라클 데이터베이스는 자동으로 다음을 수행합니다.

1. 두 세션을 종료하기 전 300초 동안 대기
2. 한 세션에서 오류가 있는 명령문 한 개 종료
3. 두 세션에서 오류가 있는 명령문 종료
4. 기본적으로 아무런 조치를 취하지 않고 DBA에게 맡김

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- Lock 메커니즘과 Oracle의 데이터 동시성 관리 방법 설명
- Lock 충돌 모니터 및 해결



Copyright © 2010, Oracle. All rights reserved.

## 연습 9 개요: 데이터 및 동시성 관리

이 연습에서는 다음 내용을 다룹니다.

- Lock 충돌 식별
- Lock 충돌 해결



Copyright © 2010, Oracle. All rights reserved.

# 10

## 언두 데이터 관리

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- DML 및 언두 데이터 생성 설명
- 언두 데이터 모니터 및 관리
- 언두 데이터와 리두 데이터의 차이점 설명
- 언두 Retention 구성
- 언두 Retention 보장
- Undo Advisor 사용

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 언두 데이터

언두 데이터는 다음 특징을 갖습니다.

- 수정되기 전 원래 데이터의 복사본입니다.
- 데이터를 변경하는 모든 트랜잭션에 대해 캡처됩니다.
- 적어도 트랜잭션이 종료될 때까지는 보존됩니다.
- 지원하는 작업:
  - 롤백 작업
  - 읽기 일관성 query
  - Oracle Flashback Query,  
Oracle Flashback Transaction,  
및 Oracle Flashback Table
  - 실패한 트랜잭션 recovery



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 언두 데이터

프로세스로 인해 데이터베이스의 데이터가 변경되는 경우 오라클 데이터베이스는 이전 값(언두 데이터)을 저장합니다. 이때 데이터는 수정되기 전 상태로 저장됩니다. 언두 데이터를 캡처하면 커밋되지 않은 데이터를 롤백할 수 있습니다. 또한 언두는 읽기 일관성 및 Flashback Query를 지원합니다. 언두를 사용하여 트랜잭션 및 테이블을 "되감기"(Flashback)할 수도 있습니다.

읽기 일관성 query는 query가 시작된 시점의 데이터와 일치하는 결과를 제공합니다. 읽기 일관성 query가 성공하려면 원래 정보가 계속 언두 정보로 존재해야 합니다. 원래 데이터를 더 이상 사용할 수 없으면, "Snapshot too old" 오류(ORA-01555)가 발생합니다. 언두 정보를 보존하고 있는 동안 오라클 데이터베이스는 읽기 일관성 query가 충족되도록 데이터를 재구성할 수 있습니다.

Flashback Query는 어떤 목적을 가지고 과거 특정 시점의 데이터 버전을 요청합니다. 요청한 과거 특정 시점에 대한 언두 정보가 있는 경우 Flashback Query를 성공적으로 완료할 수 있습니다. Oracle Flashback Transaction은 트랜잭션 및 해당 종속 트랜잭션을 백아웃하기 위해 언두를 사용하여 보완 트랜잭션을 생성합니다. Oracle Flashback Table을 사용하면 테이블을 특정 시점으로 Recovery할 수 있습니다.

또한 언두 데이터를 사용하여 실패한 트랜잭션을 Recovery할 수 있습니다. 유저가 트랜잭션의 커밋 또는 롤백을 결정하기 전에 네트워크 오류 또는 클라이언트 컴퓨터 failure로 인해 유저 세션이 비정상적으로 종료되는 경우 트랜잭션이 실패합니다. Instance가 충돌하거나 SHUTDOWN ABORT 명령을 실행하는 경우에도 실패한 트랜잭션이 발생할 수 있습니다.

## 언두 데이터(계속)

실패한 트랜잭션이 발생하는 경우 가장 안전한 동작이 선택되며 오라클 데이터베이스는 유저가 수행한 모든 변경 사항을 되돌려서 원본 데이터를 복원합니다.

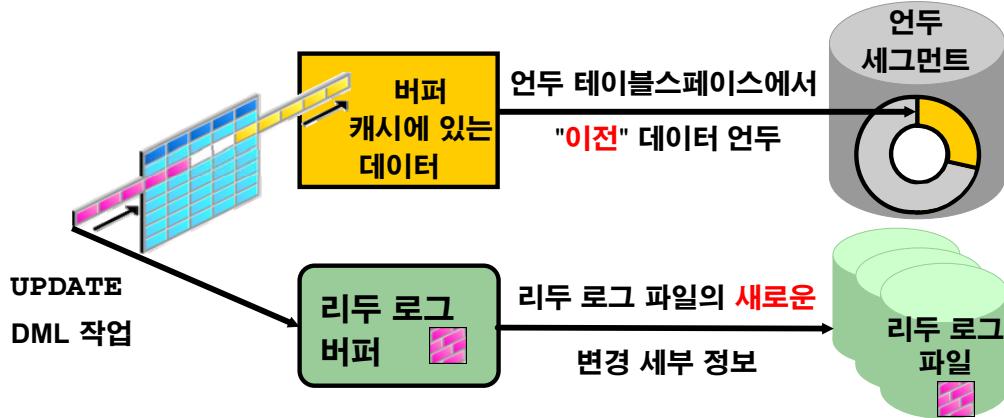
적어도 다음 중 하나에 의해 트랜잭션이 종료될 때까지 모든 트랜잭션에 대한 언두 정보를 보존합니다.

- 유저의 트랜잭션 언두(트랜잭션 롤백)
- 유저의 트랜잭션 종료(트랜잭션 커밋)
- 유저의 DDL 문 실행(CREATE, DROP, RENAME 또는 ALTER 문). 현재 트랜잭션에 DML 문이 포함되어 있는 경우 데이터베이스는 우선 트랜잭션을 커밋한 다음 DDL을 새로운 트랜잭션으로 실행 및 커밋합니다.
- 유저 세션의 비정상적인 종료(트랜잭션 롤백)
- 종료 명령에 의한 유저 세션의 정상적인 종료(트랜잭션 커밋)

보존되는 언두 데이터의 양과 보존 기간은 데이터베이스 작업량과 데이터베이스 구성에 따라 다릅니다.

**주:** Oracle Flashback Transaction은 온라인 리두 로그를 활용하여 실행을 위한 해당 언두 SQL을 마이닝합니다. 트랜잭션 시작 시간이 Flashback 트랜잭션 호출 시 제공되지 않은 경우, Oracle Flashback Transaction은 인공 시간 경계로만 언두를 사용하여 대상 트랜잭션에 대한 리두 마이닝 시작 시간을 결정합니다.

## 트랜잭션 및 언두 데이터



- 각 트랜잭션은 하나의 언두 세그먼트에만 할당됩니다.
- 하나의 언두 세그먼트는 한 번에 여러 개의 트랜잭션을 처리할 수 있습니다.

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 트랜잭션 및 언두 데이터

트랜잭션이 시작되면 해당 트랜잭션은 언두 세그먼트에 할당됩니다. 트랜잭션이 진행되는 동안 데이터가 변경되면 원래(변경되기 전) 값이 언두 세그먼트에 복사됩니다. Dynamic Performance 뷰 V\$TRANSACTION을 확인하여 어느 트랜잭션이 어느 언두 세그먼트에 할당되었는지 알 수 있습니다.

언두 세그먼트는 트랜잭션을 지원하는 데 필요한 경우 Instance에서 자동으로 생성되는 특수 세그먼트입니다. 다른 세그먼트와 마찬가지로 언두 세그먼트는 데이터 블록으로 구성된 Extent로 구성되어 있습니다. 언두 세그먼트는 필요한 경우 자동으로 확장되거나 축소(shrink)되며 할당된 트랜잭션에 대해 순환 저장 버퍼처럼 사용됩니다.

트랜잭션이 완료되거나 모든 공간이 사용될 때까지 트랜잭션은 언두 세그먼트에 Extent를 채웁니다. Extent가 가득 차서 더 많은 공간이 필요한 경우 트랜잭션은 세그먼트에 있는 다음 Extent에서 필요한 공간을 얻습니다. 모든 Extent가 사용된 후에 트랜잭션은 첫번째 Extent로 다시 래핑하거나 언두 세그먼트에 새 Extent를 할당할 것을 요청합니다.

**주:** 병렬 DML 및 DDL 작업으로 인해 실제로 트랜잭션이 두 개 이상의 언두 세그먼트를 사용할 수 있습니다. 병렬 DML 실행에 대한 자세한 사항은 *Oracle Database Administrator's Guide*를 참조하십시오.

## 언두 정보 저장

**언두 정보는 언두 세그먼트에 저장되며 언두 세그먼트는 하나의 언두 테이블스페이스에 저장됩니다. 언두 테이블스페이스의 특징은 다음과 같습니다.**

- 언두 세그먼트에만 사용됩니다.
- Recovery 시 특별한 고려 사항이 있습니다.
- 단일 instance와만 연관됩니다.
- 여러 언두 테이블스페이스 중 하나만 주어진 시간에 주어진 instance에 대해 현재 쓰기가 가능해야 합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 언두 정보 저장

언두 세그먼트는 언두 테이블스페이스라는 특수한 형식의 테이블스페이스에만 존재할 수 있습니다. 언두 테이블스페이스에 테이블 등의 다른 세그먼트 유형을 만들 수는 없습니다.

DBCA는 Small File 테이블스페이스를 자동으로 생성합니다. 또한 Big File 언두 테이블스페이스를 생성할 수도 있습니다. 그러나 짧은 동시 트랜잭션이 많은 대규모 온라인 트랜잭션 프로세싱 (OLTP) 환경에서는 파일 헤더에 경합이 발생할 수 있습니다. 여러 데이터 파일에 저장되는 언두 테이블스페이스는 이러한 잠재적인 문제를 해결할 수 있습니다.

데이터베이스에는 여러 언두 테이블스페이스가 있을 수 있지만 특정 시점에서 그 중 하나만 데이터베이스의 Instance에 대한 현재 언두 테이블스페이스로 지정될 수 있습니다.

언두 세그먼트는 자동으로 생성되며 항상 SYS가 소유합니다. 언두 세그먼트는 순환 버퍼처럼 사용되므로 각 세그먼트는 최소 두 개의 Extent를 갖습니다. 최대 Extent 수의 기본값은 데이터베이스 블록 크기에 따라 다르지만 매우 큽니다(블록 크기가 8KB인 경우 32,765개).

언두 테이블스페이스는 자동 Extent 할당을 사용하는 영구적인 로컬관리방식의 테이블스페이스로, 데이터베이스에 의해 자동으로 관리됩니다.

instance crash 등으로 인해 실패한 트랜잭션을 recovery하는 경우에는 언두 데이터가 필요하므로 언두 테이블스페이스는 instance가 MOUNT 상태일 때만 recovery할 수 있습니다. 언두 테이블스페이스에 대한 Recovery 고려 사항은 "데이터베이스 Recovery 수행" 단원에서 다릅니다.

## 언두 데이터와 리두 데이터 비교

	언두	리두
기록 내용	변경 사항을 언두하는 방법	변경 사항을 재생성하는 방법
사용 목적	롤백, 읽기 일관성, Flashback	데이터베이스 변경 사항 롤포워드
저장 위치	언두 세그먼트	리두 로그 파일
보호 대상	다중 유저 시스템에서 일관성 없는 읽기가 발생하지 않도록 보호	데이터 손실이 발생하지 않도록 보호



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 언두 데이터와 리두 데이터 비교

언두 데이터와 리두 데이터는 유사해 보이지만 사용하는 목적이 다릅니다. 언두 데이터는 변경 사항을 언두해야 하는 경우에 필요하며 읽기 일관성 및 롤백에 대해 발생합니다. 리두 데이터는 어떤 이유로 인해 데이터를 분실하여 변경 사항을 다시 수행해야 하는 경우에 필요합니다. 언두 블록 변경 사항은 리두 로그에도 기록됩니다.

커밋 프로세스는 트랜잭션의 변경 사항이 메모리가 아니라 디스크에 영구 저장되는 리두 로그 파일에 기록되었는지 검증합니다. 또한 리두 로그 파일은 대개 다중화되므로 디스크에 리두 데이터의 복사본이 여러 개 있습니다. 테이블의 블록이 실제로 저장되는 데이터 파일에 변경 사항이 아직 기록되지 않은 경우에는 지속 리두 로그에 쓰는 것만으로도 데이터베이스의 일관성이 보장됩니다.

커밋된 변경 사항이 데이터 파일에 반영되기 전에 정전이 발생하더라도 트랜잭션이 커밋되었기 때문에 문제가 발생하지 않습니다. 따라서 시스템을 다시 시작하면 정전 시 아직 데이터 파일에 반영되지 않은 리두 레코드를 룬포워드할 수 있습니다.

# 언두 관리

## 자동 언두 관리:

- 전용 언두 테이블스페이스에서 언두 데이터 및 Undo space 관리 완전 자동화
- 모든 세션에 대해 적용
- 장기 실행 query를 충족하도록 AUTOEXTEND 테이블스페이스 자체 튜닝
- Retention을 최적화하도록 고정 크기 테이블스페이스에서 자체 튜닝

## Flashback 작업을 지원하는 DBA 작업:

- 언두 Retention 구성
- 언두 테이블스페이스를 고정 크기로 변경
- 공간 및 "Snapshot too old" 오류 방지

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 언두 관리

오라클 데이터베이스는 자동 언두 관리를 통해 모든 세션에 대한 전용 테이블스페이스에서 언두 정보 및 Undo space를 완전 자동화된 방식으로 관리합니다. 시스템은 자동으로 자체 튜닝되어 언두 정보에 대한 Retention을 최적화합니다. 즉, 자동 확장 테이블스페이스를 위한 언두 Retention 기간이 장기 실행 활성 query보다 약간 길어지도록 튜닝됩니다. 고정 크기 언두 테이블스페이스의 경우 데이터베이스는 Retention을 최적화하도록 동적으로 튜닝합니다.

자동 언두 관리는 Oracle Database 11g 이상 릴리스에서 기본 제공됩니다. 수동 언두 관리는 Oracle8i 및 역호환성을 위해 지원되지만 DBA의 개입이 더 많이 요구됩니다. 수동 언두 관리 모드에서 undo space는 언두 테이블스페이스가 아닌 룰백 세그먼트를 통해 관리됩니다.

**주:** 오라클에서는 자동 언두 관리를 사용할 것을 적극 권장합니다.

기본적으로 오라클 데이터베이스는 언두 데이터 및 Undo space를 자동으로 관리하지만 데이터베이스에서 Flashback 작업을 사용하는 경우 일부 작업을 직접 수행해야 할 수도 있습니다. 언두 관리를 통해 공간 오류 및 "Snapshot too old" 오류가 발생하지 않는 것은 물론 공간이 지나치게 많이 사용되지 않도록 해야 합니다.

## 언두 Retention 구성

**UNDO\_RETENTION은 이미 커밋된 언두 정보를 보존해야 하는 기간을 초 단위로 지정합니다. 이 파라미터는 다음과 같은 경우에만 설정합니다.**

- 언두 테이블스페이스에서 AUTOEXTEND 옵션이 활성화된 경우
- LOB에 대해 언두 retention을 설정할 경우
- retention을 보장하려는 경우



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 언두 Retention 구성

UNDO\_RETENTION 초기화 파라미터는 언두 Retention의 낮은 임계값을 초 단위로 지정합니다. 자동 확장 언두 테이블스페이스를 위한 최소 언두 Retention 기간은 예상되는 가장 긴 Flashback 작업으로 가능한 한 길게 설정합니다. 자동 확장 언두 테이블스페이스의 경우 시스템은 최소한 이 파라미터에 지정된 시간 동안 언두를 보관하고 query의 언두 요구 사항을 충족하도록 언두 Retention 기간을 자동으로 튜닝합니다. 그러나 이처럼 자동 튜닝된 Retention 기간은 Flashback 작업을 수행하는 데 충분하지 않을 수도 있습니다.

고정 크기 언두 테이블스페이스의 경우 시스템은 언두 테이블스페이스 크기 및 사용 기록에 따라 최적의 언두 Retention 기간을 자동으로 튜닝하고 Retention 보장이 활성화되지 않은 경우 UNDO\_RETENTION을 무시합니다. 자동 언두 관리의 경우 UNDO\_RETENTION 설정은 슬라이드에 나열된 세 가지 경우에 사용됩니다. 이러한 세 가지 경우가 아니면 이 파라미터는 무시됩니다.

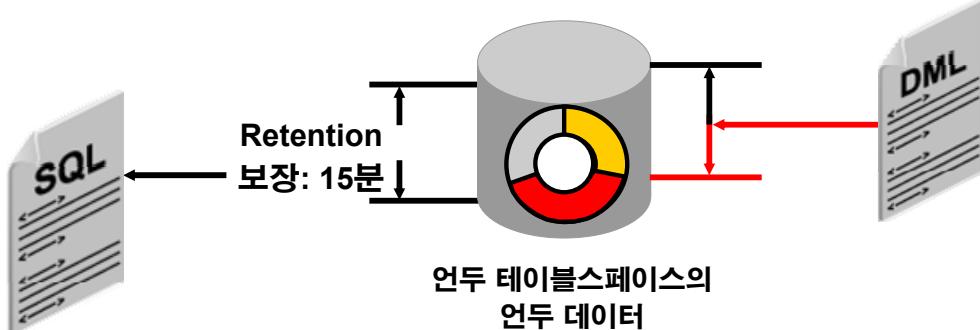
## 언두 Retention 구성(계속)

언두 정보는 다음의 세 가지 범주로 나뉩니다.

- **커밋되지 않은 언두 정보(활성):** 현재 실행 중인 트랜잭션을 지원하며 유저가 롤백하려는 경우나 트랜잭션이 실패한 경우에 필요합니다. 커밋되지 않은 언두 정보는 겹쳐쓰이지 않습니다.
- **커밋된 언두 정보(만료되지 않음):** 실행 중인 트랜잭션을 지원하는 데는 더 이상 필요하지 않지만 언두 Retention 간격을 만족시키는 데에는 여전히 필요합니다. "만료되지 않은" 언두 정보라고도 합니다. 커밋된 언두 정보는 활성 트랜잭션이 공간 부족으로 인해 실패하는 일이 없도록 하는 한도 내에서 보존됩니다.
- **만료된 언두 정보(만료됨):** 실행 중인 트랜잭션을 지원하는 데 더 이상 필요하지 않습니다. 만료된 언두 트랜잭션은 활성 트랜잭션에서 공간이 필요하면 겹쳐쓰입니다.

# 언두 Retention 보장

```
SQL> ALTER TABLESPACE undotbs1 RETENTION GUARANTEE;
```



15분 이하로 실행되는  
SELECT 문은  
항상 충족됩니다.

사용 가능한 공간보다  
더 많은 언두를 생성하는  
트랜잭션은 실패 합니다.

주: 이 예제는 900초(15분)의 UNDO\_RETENTION 설정을 기반으로 합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 언두 Retention 보장

기본적으로 언두 작업은 undo space 부족으로 인해 활성 트랜잭션이 실패하도록 허용하기 보다는 아직 만료되지 않은 커밋된 트랜잭션의 언두 정보를 겹쳐씁니다.

이 작업은 Retention을 보장하여 변경할 수 있습니다. retention을 보장하면 언두 retention 설정으로 인해 트랜잭션이 실패하더라도 언두 retention 설정을 시행합니다.

RETENTION GUARANTEE는 초기화 파라미터라기보다는 테이블스페이스 속성입니다. 이 속성은 SQL 명령행 문을 통해서만 변경할 수 있습니다. retention을 보장하도록 언두 테이블스페이스를 변경하는 구문은 다음과 같습니다.

```
SQL> ALTER TABLESPACE undotbs1 RETENTION GUARANTEE;
```

보장된 언두 테이블스페이스를 일반 설정으로 되돌리려면 다음 명령을 사용하십시오.

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

retention 보장은 언두 테이블스페이스에만 적용됩니다. 비언두 테이블스페이스에 retention 보장을 설정하려고 하면 다음 오류가 발생합니다.

```
SQL> ALTER TABLESPACE example RETENTION GUARANTEE;  
ERROR at line 1:  
ORA-30044: 'Retention' can only specified for undo tablespace
```

# 언두 테이블스페이스를 고정 크기로 변경

## 이유:

- Flashback 작업 지원
- 테이블스페이스 증가 제한

## 워크플로우:

1. 일반 작업 로드를 실행합니다.
2. 자체 튜닝 메커니즘을 통해 필요한 최소 크기가 설정됩니다.
3. (선택 사항) Undo Advisor를 사용하여 향후 증가되는 규모에 필요한 크기를 계산합니다.
4. (선택 사항) 언두 테이블스페이스를 고정 크기로 변경합니다.



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## 언두 테이블스페이스를 고정 크기로 변경

언두 테이블스페이스를 고정 크기로 변경하는 데는 두 가지 이유가 있습니다. 첫번째는 향후 언두가 사용될 것으로 예상되는 Flashback 작업을 지원하기 위해서이고 두번째는 테이블스페이스가 지나치게 증가하는 것을 막기 위해서입니다.

언두 테이블스페이스를 고정 크기로 변경하려는 경우 크기를 충분하게 설정하지 않으면 다음과 같은 오류가 발생할 수 있습니다.

- DML Failure(새 트랜잭션에 대한 Undo space가 부족하기 때문)
- "Snapshot too old" 오류(읽기 일관성을 유지하기 위한 언두 데이터가 부족하기 때문)

오라클에서는 언두 테이블스페이스가 필요한 최소 크기로만 증가할 수 있도록 전체 작업 로드를 주기적으로 실행할 것을 권장합니다. 자동으로 수집되는 통계에는 가장 길게 실행되는 query의 기간 및 언두 생성 비율이 포함됩니다. Flashback 작업이 없는 시스템은 물론 향후 장기 실행 query가 예상되지 않는 시스템의 경우 이러한 통계를 바탕으로 최소 언두 테이블스페이스 크기를 계산하면 도움이 됩니다.

Undo Advisor를 사용하면 장기 실행 query 및 Flashback의 언두 기간에 대해 원하는 기간을 입력할 수 있습니다.

# 일반 언두 정보

Database Instance: orcl.oracle.com >

**Automatic Undo Management**

In the General tab, you can view the current undo settings for your instance and use the Undo Advisor to analyze the undo tablespace requirements. This analysis can be performed based on the specified analysis period or the desired undo retention. The system activity for the specified time period can be viewed in the System Activity tab.

**General      System Activity**

**Undo Retention Settings**

Undo Retention (minutes)	15
Retention Guarantee	No

**Undo Tablespace for this Instance**

Tablespace	UNDOTBS1
Size (MB)	100
Auto-Extensible	Yes

**현재 테이블스페이스 크기**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 일반 언두 정보

Enterprise Manager에서 Server > Automatic Undo Management를 선택합니다.

General과 System Activity라는 두 개의 페이지가 있는데 General 페이지의 윗부분에는 Undo Retention Settings 및 이 Instance의 언두 테이블스페이스에 대한 정보가 표시됩니다.

# Undo Advisor 사용

**Undo Advisor: Undo Retention and Undo Tablespace Sizing Advice**

Undo retention is the length of time that undo data is retained in the undo tablespaces. Undo data must be retained for the length of the longest running query, the longest running transaction, and the longest flashback duration (except for Flashback Database). The undo tablespace should be sized large enough to hold the undo generated by the database during the undo retention period. Note that the undo retention parameter is also used as the retention value for LOB columns.

**Analysis Period**

Analysis Time Period: Last One Hour

Desired Undo Retention:

- Automatically chosen based on longest query in analysis period
- Specified manually to allow for longer duration queries or flashback

Duration: minutes

**Run Analysis**

**Analysis Results**

Selected Analysis Time Period: Jun 17, 2009 11:04:47 AM GMT+07:00 To Jun 17, 2009 11:04:47 PM GMT+07:00

Minimum Required Undo Tablespace Size (MB): 10

Recommended Undo Tablespace Size (MB): 15

**TIP** Recommended size is three times the minimum size to allow for workload fluctuations.

Potential Problems: No Problem Found  
Recommendations: No Recommendation

**Show Graph**

**Required Tablespace Size (MB)** vs **Undo Retention (minutes)**

**TIP** Clicking a point on the graph updates the Duration field and the Minimum Required Undo Tablespace Size field.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

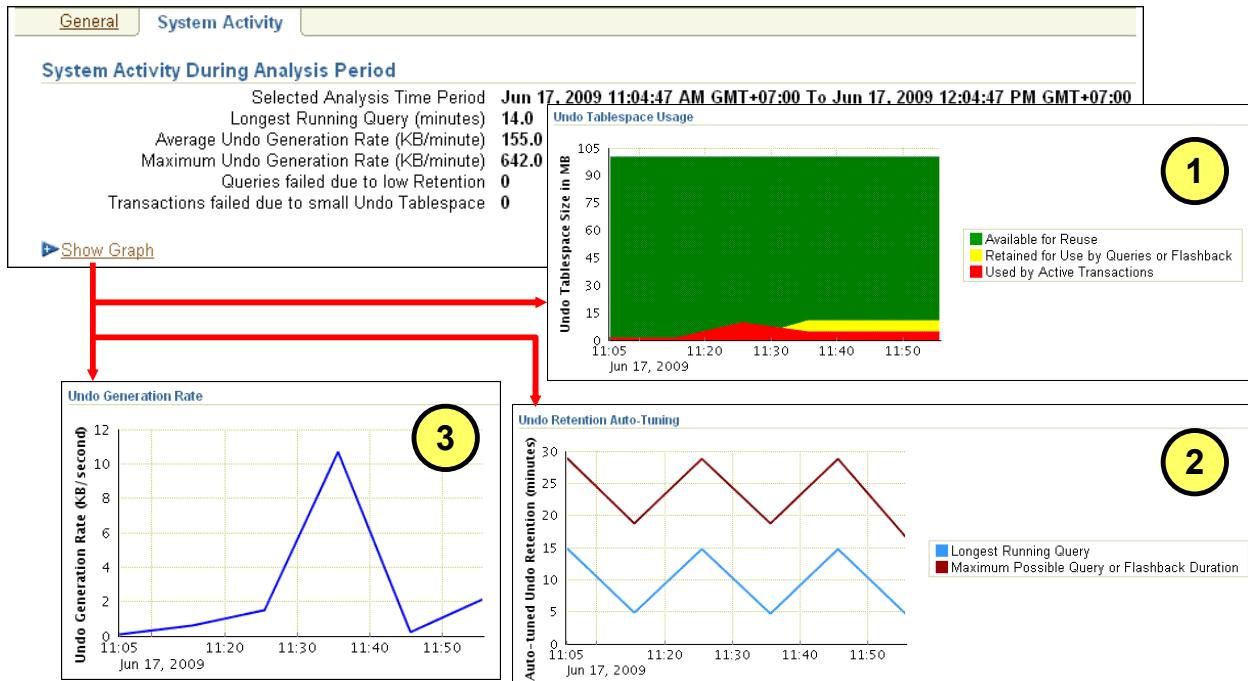
## Undo Advisor 사용

General 언두 페이지의 중간 부분에서는 Undo Advisor에 액세스할 수 있습니다. 또한 지정된 언두 retention을 만족시키는 데 필요한 언두 테이블스페이스 크기를 예측하여 제공합니다.

Undo Advisor의 Analysis 영역에는 Retention 기간을 지원하는 데 필요한 테이블스페이스 크기가 표시됩니다. 그래프의 한 지점을 눌러 선택한 기간을 지원하는 데 필요한 테이블스페이스 크기를 확인할 수 있습니다.

Edit Undo Tablespace 버튼을 누른 다음 Datafile 섹션의 Edit를 눌러 언두 테이블스페이스를 고정 크기로 변경합니다.

# 시스템 작업 보기



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 시스템 작업 보기

페이지의 윗부분에는 선택한 기간 동안의 시스템 작업이 표시됩니다.

그 아래에는 다음과 같은 세 개의 그래프가 나타납니다.

1. **Undo Tablespace Usage:** 테이블스페이스 크기(MB 단위)를 그 달의 일별로 표시합니다.
2. **Undo Retention Auto-Tuning:** 튜닝된 언두 Retention(분 단위)을 그 달의 일별로 시각화합니다.
3. **Undo Generation Rate:** 언두 생성(초당 KB 단위)을 그 달의 일별로 표시합니다.

## 퀴즈

15분 미만의 모든 query에서 읽기 일관성에 필요한 언두 데이터를 찾을 수 있도록 보장하기 위해 필요한 작업은 UNDO\_RETENTION 파라미터를 15분으로 설정을 하는 것입니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 퀴즈

언두 데이터와 관련 없는 내용은 무엇입니까?

1. 변경 사항을 언두하는 방법에 대한 레코드 제공
2. 롤백, 읽기 일관성 및 flashback에 사용
3. 메모리에만 저장되고 디스크에 기록되지 않음
4. 다중 유저 시스템에서 일관성 없는 읽기가 발생하지 않도록 보호

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 3

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- DML 및 언두 데이터 생성 설명
- 언두 데이터 모니터 및 관리
- 언두 데이터와 리두 데이터의 차이점 설명
- 언두 Retention 구성
- 언두 Retention 보장
- Undo Advisor 사용



Copyright © 2010, Oracle. All rights reserved.

## 연습 10 개요: 언두 세그먼트 관리

이 연습에서는 다음 내용을 다룹니다.

- 시스템 작업 보기
- 48시간 retention 간격을 지원하도록 언두 테이블스페이스 크기를 계산
- 48시간 retention 간격을 지원하도록 언두 테이블스페이스 수정



Copyright © 2010, Oracle. All rights reserved.



# 11 오라클 데이터베이스 감사(audit) 구현

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 보안 및 감사(audit)를 위한 DBA 책임 설명
- 표준 데이터베이스 감사(Audit) 활성화
- 감사(Audit) 옵션 지정
- 감사(Audit) 정보 검토
- Audit Trail 유지 관리



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

이 단원부터는 Oracle 보안에 대해 알아봅니다. 다음 설명서에서 추가 정보가 제공됩니다.

- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*
- *Oracle Database Security Guide*

다음 과정에서 추가 교육이 제공됩니다.

- *Oracle Database 11g: Administration Workshop II*
- *Oracle Database 11g: Security*

## 책임 구분

- DBA 권한을 가진 유저는 신뢰되어야 합니다.
  - 신뢰의 남용
  - Audit Trail로 신뢰된 지위 보호
- DBA 책임을 공유해야 합니다.
- 계정을 공유해서는 안됩니다.
- DBA 및 시스템 관리자는 서로 다른 사람이어야 합니다.
- 운영자와 DBA의 책임을 구분합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 책임 구분

다음은 책임 구분을 충족하기 위한 주요 요구 사항입니다.

**DBA가 신뢰되어야 함:** DBA를 제한하기는 어렵습니다. DBA는 작업을 수행하기 위해 높은 레벨의 권한이 필요합니다. DBA는 신뢰된 지위의 사람이어야 하며 철저히 검증되어야 합니다. 신뢰된 DBA라도 책임감을 지녀야 합니다. 다음 사항을 고려하십시오.

- **신뢰의 남용:** DBA는 DBA\_USERS 뷰에서 암호화된 암호(encrypted password)를 오용할 가능성이 있습니다.
- **Audit Trail로 신뢰된 지위 보호:** 감사(Audit)가 신중히 시행되고 지침을 따르는 경우 Audit Trail을 통해 특정 사람이 절차를 위반하거나 잘못된 행동을 하지 않았음을 입증할 수 있습니다. 악의적인 유저가 신뢰된 유저에게 협의를 두는 경우 잘 구성된 Audit Trail을 통해 기록을 추적할 수 있습니다.

**Oracle Database Vault:** 데이터베이스가 책임 구분을 적용해야 하는 상황 또는 DBA가 일부 또는 전체 데이터베이스 스키마의 데이터를 볼 수 없는 상황에서는 Oracle Database Vault 옵션을 사용할 수 있습니다.

# 데이터베이스 보안

**보안 시스템은 내부 데이터의 기밀성을 보장합니다. 보안에는 다음과 같은 측면이 있습니다.**

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



**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 보안

Oracle Database 11g는 업계 최고의 보안 시스템용 프레임워크를 제공합니다. 그러나 프레임워크가 효율적으로 작동하려면 데이터베이스 관리자가 최적의 사용법에 따라 지속적으로 데이터베이스 작업을 모니터해야 합니다.

### 데이터 및 서비스에 대한 액세스 제한

모든 유저가 모든 데이터에 액세스할 수 있도록 해서는 안됩니다. 데이터베이스에 저장된 내용에 따라 다르지만 보통 업무 요구 사항, 고객의 기대치 및 법적 제재에 따라 제한된 액세스 권한을 부여할 수 있습니다. 신용 카드 정보, 의료 정보 데이터, ID 정보 등에는 권한 없이 액세스하지 못하도록 보호해야 합니다. 오라클 데이터베이스는 매우 상세한 권한 부여 제어를 제공하여 데이터베이스 액세스를 제한합니다. 액세스 제한에는 최소 권한의 원칙 적용이 포함되어야 합니다.

## 데이터베이스 보안(계속)

### 유저 인증

중요한 데이터에 대한 액세스를 제어하려면 먼저 시스템에서 누가 데이터에 액세스하려고 하는지 확인해야 합니다. 인증 정보가 노출되면 다른 모든 보안 예방책은 소용 없게 됩니다. 가장 기본적인 형태의 유저 인증은 유저에게 암호와 같이 유저가 알고 있는 정보를 제공하도록 하는 것입니다. 암호 지정 시 간단한 규칙을 따르도록 하면 시스템 보안을 크게 향상시킬 수 있습니다. 좀 더 강력한 인증 방법은 유저에게 토큰이나 공용 키 기반 구조(PKI) 인증서 등을 제공하도록 하는 것입니다. 훨씬 더 강력한 형태의 인증은 지문, 홍채 스캔, 뼈 구조 형태 등 고유한 생물 통계학적(Biometrics) 특징을 통해 유저를 식별하는 것입니다. 오라클 데이터베이스는 Advanced Security Option을 통해 토큰 기반, 생물 통계학적 기반 및 인증서 기반 식별과 같은 고급 인증 기법을 지원합니다. 사용하지 않는 유저 계정은 인증이 노출되지 않도록 lock해 두어야 합니다.

### 의심스러운 작업 모니터

권한이 부여되었으며 인증된 유저도 시스템을 손상시킬 수 있습니다. 예를 들어, 어떤 직원이 신용 카드 정보, 조사 결과 또는 다른 중요한 정보를 대량으로 query하기 시작하면 이와 같은 비정상적인 데이터베이스 작업을 식별하는 것이 정보 도용 감지의 첫걸음이 될 수 있습니다. 오라클 데이터베이스는 유저 작업을 추적하고 의심이 가는 경향을 식별할 수 있도록 다양한 감사(Audit) 도구를 제공합니다.

## 준수 여부 모니터

모니터 또는 감사(Audit)는 보안 프로시저의 필수 요소입니다.

다음 감사(Audit)를 검토합니다.

- 필수 감사(Audit)
- 표준 데이터베이스 감사(Audit)
- 값 기준 감사(Audit)
- FGA(Fine-grained auditing)
- SYSDBA(및 SYSOPER) 감사(Audit)



ORACLE®

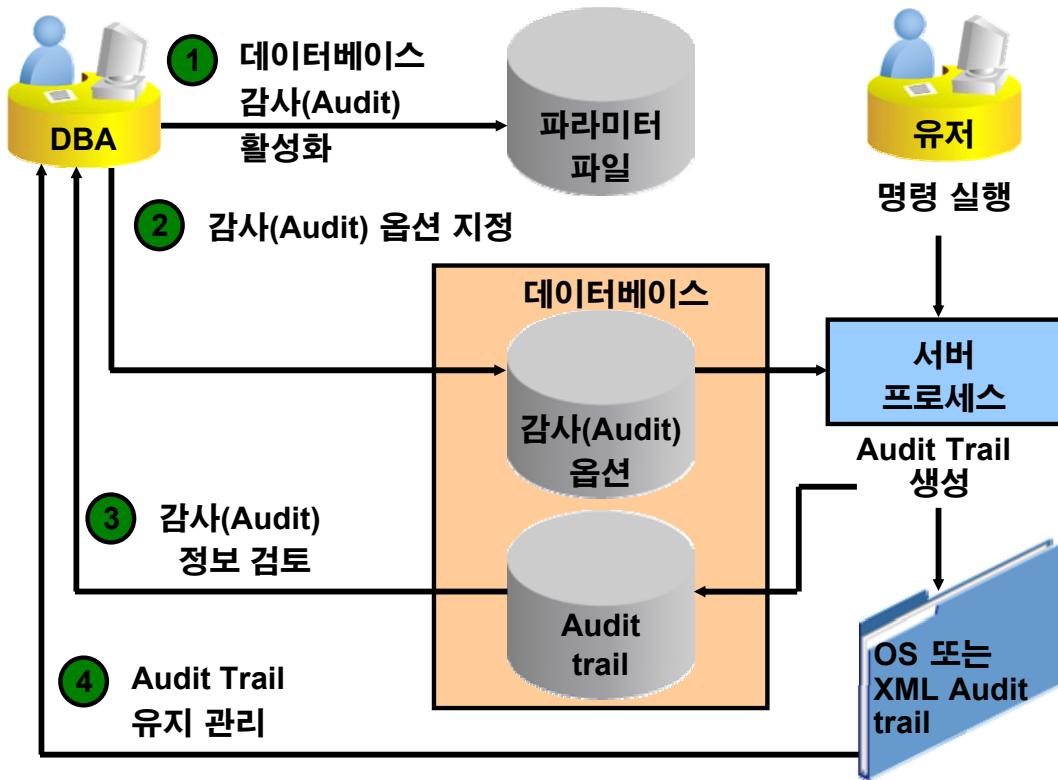
Copyright © 2010, Oracle. All rights reserved.

### 준수 여부 모니터

감사(Audit) 과정에서는 시스템에서 발생하는 일에 대한 정보를 캡처하고 저장하게 되어 시스템에서 처리할 작업량이 증가하므로 원하는 이벤트만 캡처되도록 감사(Audit)를 제한해야 합니다. 적절하게 감사(Audit)를 제한하면 시스템 성능에 미치는 영향을 최소화할 수 있습니다. 그렇지 않은 경우 시스템 성능에 큰 영향을 미칠 수 있습니다.

- **필수 감사(Audit):** 모든 오라클 데이터베이스는 다른 감사(Audit) 옵션이나 파라미터에 관계없이 특정 작업을 감사(Audit)합니다. 필수 감사(Audit) 로그가 필요한 이유는 데이터베이스가 권한이 있는 유저의 연결과 같은 일부 데이터베이스 작업을 기록할 필요가 있기 때문입니다.
- **표준 데이터베이스 감사(Audit):** AUDIT\_TRAIL 초기화 파라미터를 사용하여 시스템 레벨에서 활성화됩니다. 감사(Audit)를 활성화하고 나면 감사(Audit)할 객체 및 권한을 선택하고 AUDIT 명령을 사용하여 감사(Audit) 속성을 설정합니다.
- **값 기준 감사(Audit):** 표준 데이터베이스 감사(Audit)의 확장된 감사(Audit) 기능이며, 감사(Audit)한 이벤트 발생 사실뿐만 아니라 삽입, 갱신 또는 삭제된 실제 값을 캡처합니다. 값 기준 감사(Audit)는 데이터베이스 트리거를 통해 구현됩니다.
- **FGA(Fine-Grained Auditing):** 표준 데이터베이스 감사(Audit)의 확장된 감사(Audit) 기능이며, 이벤트 발생 사실만이 아니라 실행된 실제 SQL 문을 캡처합니다.
- **SYSDBA(및 SYSOPER) 감사(Audit):** DBA와 운영 체제 Audit Trail에서 DBA 작업을 모니터하는 감사자 또는 보안 관리자 간에 감사(Audit) 책임을 구분합니다.

## 표준 데이터베이스 감사(Audit)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 표준 데이터베이스 감사(Audit)

데이터베이스 감사(Audit)를 사용하려면 먼저 정적 AUDIT\_TRAIL 파라미터가 감사(Audit) 레코드의 저장 영역 위치를 가리키도록 설정해야 합니다. 그러면 데이터베이스 감사(Audit)가 활성화됩니다. 데이터베이스 감사(Audit)를 활성화하고 감사(Audit) 옵션(로그인 이벤트, 시스템 및 객체 권한 실행 또는 SQL 문 사용)을 지정하면 데이터베이스가 감사(Audit) 정보를 수집하기 시작합니다.

AUDIT\_TRAIL을 OS로 설정하면 감사(Audit) 레코드가 운영 체제의 감사(Audit) 시스템에 저장됩니다. 이러한 감사(Audit) 레코드는 Windows 환경에서는 이벤트 로그에, UNIX 또는 Linux 환경에서는 AUDIT\_FILE\_DEST 파라미터로 지정되는 파일에 저장됩니다.

AUDIT\_TRAIL 파라미터가 DB 또는 DB, EXTENDED로 설정된 경우 SYS 스키마의 일부인 DBA\_AUDIT\_TRAIL 뷰에서 감사(Audit) 레코드를 검토할 수 있습니다.

AUDIT\_TRAIL이 XML 또는 XML,EXTENDED로 설정된 경우 AUDIT\_FILE\_DEST 파라미터가 가리키는 디렉토리의 XML 파일에 감사(Audit) 레코드가 기록됩니다.

V\$XML\_AUDIT\_TRAIL 뷰를 사용하면 이 디렉토리의 모든 XML 파일을 볼 수 있습니다.

Audit Trail을 유지 관리하는 것은 중요한 관리 작업입니다. 감사(Audit) 옵션 포커스에 따라 Audit Trail이 매우 빠르고 크게 확장될 수 있습니다. 잘못 유지 관리하면 Audit Trail이 너무 많은 레코드를 생성하여 시스템 성능에 영향을 미칠 수 있습니다. 감사(Audit) 오버헤드는 생성되는 레코드 수에 직접적으로 관련되어 있습니다.

# Audit Trail 구성

## AUDIT\_TRAIL을 사용하여 데이터베이스 감사(Audit) 활성화

The screenshot shows the Oracle Enterprise Manager Initialization Parameters page for the database instance 'orcl.oracle.com'. The 'SPFILE' tab is selected. A yellow callout box highlights the 'audit\_trail' parameter, which is set to 'XML'. A legend on the right lists the audit trail levels: NONE, OS, DB, DB\_EXTENDED, XML, and XML\_EXTENDED. Below the table, a command is shown to alter the system parameter:

```
ALTER SYSTEM SET AUDIT_TRAIL='XML' SCOPE=SPFILE;
```

이 정적 초기화 파라미터를 수정한 후에 데이터베이스를 재시작합니다.

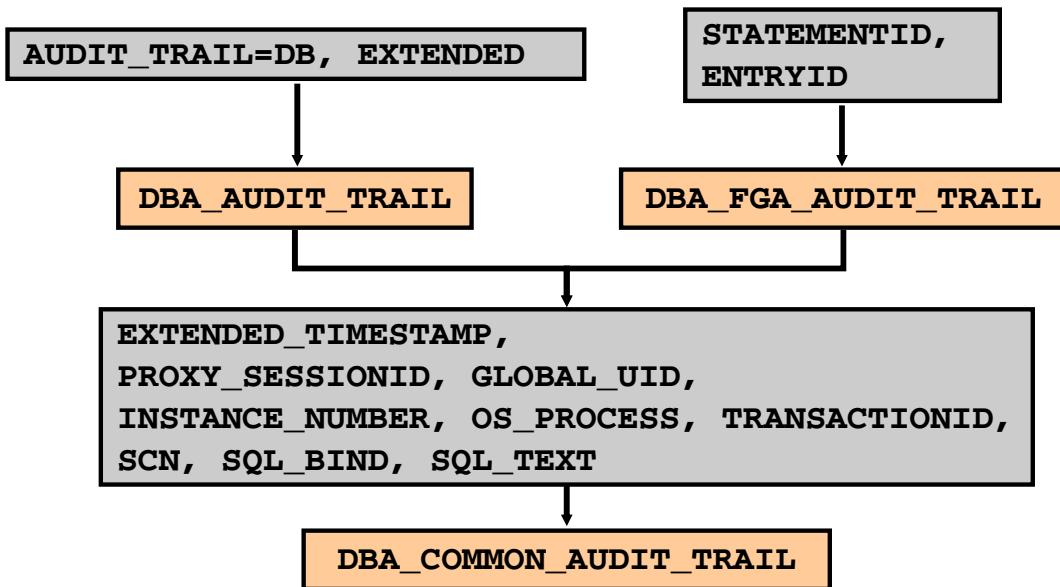
ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Audit Trail 구성

Enterprise Manager(Initialization Parameters 페이지) 또는 SQL\*Plus의 ALTER SYSTEM SET 명령을 사용하여 AUDIT\_TRAIL 파라미터를 설정할 수 있습니다. 이 파라미터는 정적 파라미터이므로 변경 사항을 적용하려면 데이터베이스를 재시작해야 합니다. DBCA(Database Configuration Assistant)를 사용하여 데이터베이스를 생성한 경우 기본적으로 audit\_trail 파라미터가 DB로 설정됩니다. AUDIT\_TRAIL이 DB로 설정된 경우 기본 동작은 Audit Trail을 데이터베이스 AUD\$ 테이블에 기록하는 것입니다. 이 감사(Audit)는 대부분의 사이트에서 데이터베이스 성능에 큰 영향을 주지 않아야 합니다. 오라클은 OS Audit Trail 파일을 사용할 것을 권장합니다. 데이터베이스를 수동으로 생성한 경우(CREATE DATABASE 명령 사용), AUDIT\_TRAIL이 기본적으로 NONE으로 설정됩니다.

# Audit Trail 군일화



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Audit Trail 군일화

오라클 데이터베이스는 표준 및 Fine-Grained Auditing에 대해 동일한 필드를 추적하므로 데이터베이스 작업을 쉽게 분석할 수 있습니다. 이렇게 하기 위해, 표준 Audit Trail과 Fine-Grained Audit Trail은 서로를 보완하는 속성을 가지고 있습니다.

표준 감사(Audit)를 통해 수집되는 추가 정보에는 다음이 포함됩니다.

- 시스템의 모든 변경 사항을 기록하는 SCN(시스템 변경 번호)
- 유저가 실행한 정확한 SQL 텍스트 및 이 SQL 텍스트와 함께 사용된 바인드 변수.  
이 열들은 AUDIT\_TRAIL=DB, EXTENDED를 지정한 경우에만 표시됩니다.

Fine-Grained Auditing를 통해 수집되는 추가 정보에는 다음이 포함됩니다.

- 각 감사(Audit) 레코드에 대한 일련 번호
- 단일 명령문으로 발생되는 다중 감사(Audit) 항목에 연결되는 명령문 번호

공통 속성에는 다음이 포함됩니다.

- UTC(협정 세계 표준시)로 표시된 세계 시간 기록. 이 필드는 지리적으로 다른 위치, 다른 시간대에 있는 여러 서버를 모니터할 때 유용합니다.
- 각 RAC(Real Application Clusters) instance에 대해 고유한 instance 번호
- 단일 트랜잭션의 여러 감사(Audit) 레코드를 그룹화할 수 있는 트랜잭션 식별자

DBA\_COMMON\_AUDIT\_TRAIL 뷰는 표준 감사(Audit) 로그 레코드와 Fine-Grained Audit 로그 레코드를 결합합니다.

## 감사(Audit) 옵션 지정

- SQL 문 감사(Audit):

```
AUDIT table;
```

- 시스템 권한 감사(Audit)(제한 및 비제한):

```
AUDIT select any table, create any trigger;
AUDIT select any table BY hr BY SESSION;
```

- 객체 권한 감사(Audit)(제한 및 비제한):

```
AUDIT ALL on hr.employees;
AUDIT UPDATE,DELETE on hr.employees BY ACCESS;
```

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 감사(Audit) 옵션 지정

**SQL 문 감사(Audit):** 슬라이드에 표시된 문은 CREATE TABLE, DROP TABLE, TRUNCATE TABLE 등을 포함하여 테이블에 영향을 주는 모든 DDL(데이터 정의어) 문을 감사(Audit)할 수 있습니다. username 또는 Success/Failure를 사용하여 SQL 문 감사(Audit)를 제한할 수 있습니다.

SQL> AUDIT TABLE BY hr WHENEVER NOT SUCCESSFUL;

**시스템 권한 감사(Audit):** DROP ANY TABLE과 같은 시스템 권한 행사를 감사(Audit)하는 데 사용되며 username 또는 Success/Failure를 사용하여 제한할 수 있습니다. 기본적으로 감사(Audit)는 BY ACCESS입니다. 감사(Audit)한 시스템 권한을 행사할 때마다 감사(Audit) 레코드가 생성됩니다. BY SESSION 절을 사용하여 세션별로 하나의 레코드만 생성되도록 해당 레코드 그룹화할 수 있습니다. 이 방법으로 유저가 다른 유저가 소유한 테이블에 대해 여러 개의 UPDATE 문을 실행할 경우 하나의 감사(Audit) 레코드만 수집할 수 있습니다. 시스템 권한 감사(Audit)가 성능 및 저장 영역에 미치는 영향을 제한하려면 BY SESSION 절을 사용하는 것이 좋습니다.

**객체 권한 감사(Audit):** 테이블, 뷰, 프로시저, 시퀀스, 디렉토리 및 유저 정의 데이터 유형에 대한 작업을 감사(Audit)하는 데 사용할 수 있습니다. 이 유형의 감사(Audit)는 Success/Failure를 사용하여 제한할 수 있으며 세션 또는 액세스별로 그룹화할 수 있습니다. 시스템 권한 감사(Audit)와 달리 기본적으로 세션별로 그룹화됩니다. 각 작업마다 Audit Trail 레코드가 별도로 생성되도록 하려면 BY ACCESS를 명시적으로 지정해야 합니다.

# 기본 감사(Audit)

기본적으로 감사(Audit)되는 권한		
ALTER ANY PROCEDURE	CREATE ANY LIBRARY	GRANT ANY PRIVILEGE
ALTER ANY TABLE	CREATE ANY PROCEDURE	GRANT ANY ROLE
ALTER DATABASE	CREATE ANY TABLE	DROP ANY PROCEDURE
ALTER PROFILE	CREATE EXTERNAL JOB	DROP ANY TABLE
ALTER SYSTEM	CREATE PUBLIC DATABASE LINK	DROP PROFILE
ALTER USER	CREATE SESSION	DROP USER
AUDIT SYSTEM	CREATE USER	EXEMPT ACCESS POLICY
CREATE ANY JOB	GRANT ANY OBJECT PRIVILEGE	
기본적으로 감사(Audit)되는 명령문		
SYSTEM AUDIT BY ACCESS ROLE BY ACCESS		

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 기본 감사(Audit)

Oracle Database 11g에서 감사(Audit)가 활성화된 경우 보안에 매우 중요한 명령문 및 특정 권한이 기본적으로 감사(Audit)됩니다. 이러한 권한 및 명령문은 슬라이드에 나열되어 있고 성공 및 Failure 시 액세스를 기준으로 모든 유저에 대해 감사(Audit)됩니다.

# Enterprise Manager Audit 페이지

**Audit Settings**

**Configuration**

Audit Trail DB  
 Audit SYS User Operations FALSE  
 Audit File Directory /u01/app/oracle/admin/orcl/adump  
Audit File Directory value is effective only when Audit Trail is set to "OS" or "XML".

Default Options For Future Audited Objects 0

**Audit Trails**

Database Audit Trail Audited Failed Logins  
Audited Privileges  
Audited Objects  
 Operating System Audit Trail View OS Audit Trails

Privilege	Select	Privilege	User	Proxy	Success	Failure
	<input type="checkbox"/>	DROP PROFILE			BY ACCESS	BY ACCESS
	<input type="checkbox"/>	ALTER ANY TABLE			BY ACCESS	BY ACCESS
	<input type="checkbox"/>	ALTER SYSTEM			BY ACCESS	BY ACCESS
	<input type="checkbox"/>	ALTER DATABASE			BY ACCESS	BY ACCESS
	<input type="checkbox"/>	DROP USER			BY ACCESS	BY ACCESS

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## Enterprise Manager Audit 페이지

Database Control Home 페이지에서 Server 탭을 누른 다음 Security 영역에서 Audit Settings 링크를 누르면 Audit 페이지로 이동할 수 있습니다.

Audit 페이지에는 다음 영역이 포함되어 있습니다.

- **Configuration:** 현재 구성 파라미터 값을 보여주며 파라미터 값을 편집할 수 있는 링크가 있습니다.
- **Audit Trails:** 수집된 감사(Audit) 정보에 쉽게 액세스할 수 있도록 합니다.

다음 탭 페이지에서 감사(Audit) 옵션을 설정/해제할 수 있습니다.

- **Audited Privileges:** 감사(Audit)되는 권한을 보여줍니다.
- **Audited Objects:** 감사(Audit)되는 객체를 보여줍니다.
- **Audited Statements:** 감사(Audit)되는 명령문을 보여줍니다.

# 감사(Audit) 정보 사용 및 유지 관리

**Audit Trails**

- Database Audit Trail [Audited Failed Logins](#)
- [Audited Privileges](#)
- Audited Objects** (highlighted with a red box)
- Operating System Audit Trail [View Or Audit Trails](#)

**Audited Objects**

[Filter Result](#) [Return](#)

[Hide SQL](#)

```
SELECT 'OWNER', 'OBJ_NAME', 'USERNAME', 'ACTION_NAME', 'TIMESTAMP' FROM 'SYS'.'DBA_AUDIT_OBJECT'
ORDER BY extended_timestamp desc
```

Schema	Object Name	User Name	Action	Time
INVENTORY	PRODUCT_MASTER	DBA1	ALTER TABLE	2008-08-13 22:47:56.0
INVENTORY	PRODUCT_ON_HAND	DBA1	CREATE TABLE	2008-08-13 16:45:49.0

[Previous 25](#) 26-34 of 34 [Next](#)

감사(Audit) 옵션을 사용하지 않는 경우 비활성화합니다.

**Confirmation**

Are you sure you want to remove the 4 selected audited objects?

The audited statements you remove will no longer be audited on the objects.

[Hide SQL](#)

```
NOAUDIT COMMENT ON HR.EMPLOYEES
NOAUDIT INDEX ON HR.EMPLOYEES
NOAUDIT LOCK ON HR.EMPLOYEES
NOAUDIT RENAME ON HR.EMPLOYEES
```

[No](#) [Yes](#)

ORACLE®

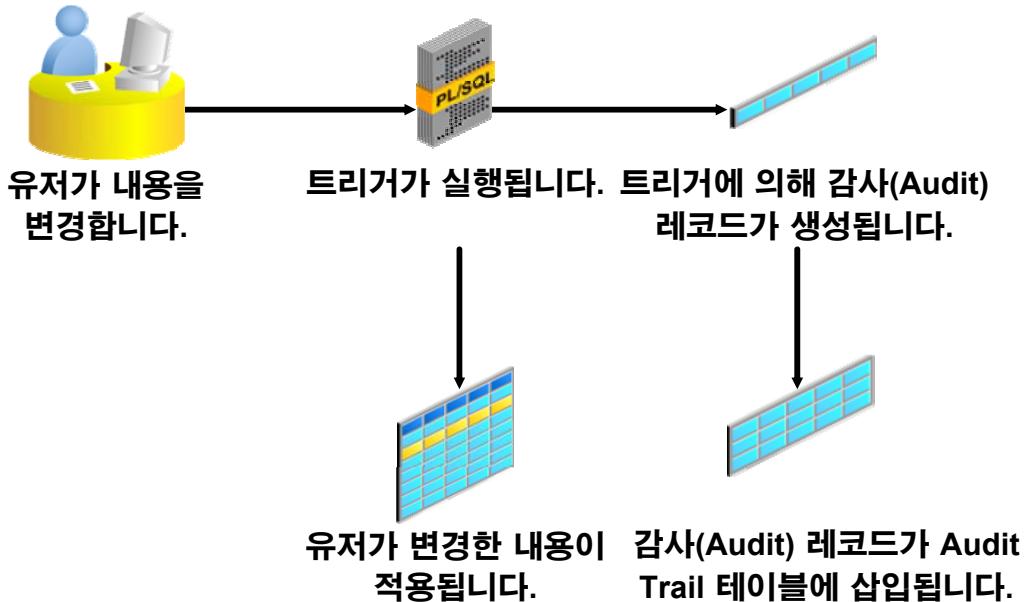
Copyright © 2010, Oracle. All rights reserved.

## 감사(Audit) 정보 사용 및 유지 관리

### 최적의 사용법(Best Practice Tip)

감사(Audit)를 수행하면 Audit Trail에 대한 쓰기 횟수에 비례하여 성능이 저하됩니다. 사이트의 요구 사항에 맞게 감사(Audit) 옵션을 조정하려면 보안 정책을 충족하는 데 필요한 옵션만 활성화하십시오. 그리고 Audit Trail 항목 수를 줄이도록 감사(Audit)를 제한해야 합니다.

## 값 기준 감사(Audit)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 값 기준 감사(Audit)

데이터베이스 감사(Audit)는 감사(Audit)된 객체에서 삽입, 갱신 및 삭제가 발생했다는 사실을 기록하지만 이렇게 변경된 실제 값을 캡처하지는 않습니다. 데이터베이스 감사(Audit) 기능을 확장하기 위해 값 기준 감사(Audit)는 데이터베이스 트리거(이벤트 방식 PL/SQL 생성자)를 활용하여 변경된 값을 캡처합니다.

유저가 적절한 트리거를 연결하여 테이블에서 데이터를 삽입, 갱신 또는 삭제하면 트리거는 백그라운드에서 감사(Audit) 정보를 포함하도록 설계된 테이블에 감사(Audit) 정보를 복사합니다. 값 기준 감사(Audit)는 삽입, 갱신 또는 삭제 작업이 발생할 때마다 감사(Audit) 트리거 코드를 실행해야 하므로 표준 데이터베이스 감사(Audit)에 비해 성능이 저하되는 경향이 있습니다. 성능 저하 정도는 트리거 코드의 효율성에 따라 다릅니다. 값 기준 감사(Audit)는 표준 데이터베이스 감사(Audit)를 통해 캡처한 정보가 불충분할 때만 사용해야 합니다.

값 기준 감사(Audit)는 유저 또는 Third-party 코드에 의해 구현됩니다. 오라클 데이터베이스는 값 기준 감사(Audit) 시스템을 작성할 수 있도록 하는 PL/SQL 생성자를 제공합니다.

## 값 기준 감사(Audit)(계속)

값 기준 감사(Audit)의 핵심 요소는 감사(Audit) 정보를 캡처하기 위해서 생성되는 단순한 PL/SQL 트리거인 감사(Audit) 트리거입니다.

다음은 일반적인 감사(Audit) 트리거의 예제입니다.

```
CREATE OR REPLACE TRIGGER system.hrsalary_audit
    AFTER UPDATE OF salary
    ON hr.employees
    REFERENCING NEW AS NEW OLD AS OLD
    FOR EACH ROW
BEGIN
    IF :old.salary != :new.salary THEN
        INSERT INTO system.audit_employees
        VALUES (sys_context('userenv', 'os_user'), sysdate,
                sys_context('userenv', 'ip_address'),
                :new.employee_id ||
                ' salary changed from '||:old.salary||
                ' to '||:new.salary);
    END IF;
END;
/

```

이 트리거는 hr.employees 테이블의 급여 열에 대한 변경 사항을 캡처하도록 감사(Audit)를 제한합니다. 행이 갱신되면 트리거가 급여 열을 검사합니다. 이전 급여가 새 급여와 동일하지 않으면 트리거가 SYSTEM 스키마에서 별도의 작업을 통해 생성된 audit\_employees 테이블에 감사(Audit) 레코드를 삽입합니다. 감사(Audit) 레코드는 Username, 변경된 IP 주소, 변경된 레코드를 식별하는 Primary Key 및 변경된 실제 급여 값을 포함합니다.

표준 데이터베이스 감사(Audit)를 통해 데이터를 충분히 수집할 수 없는 경우에도 데이터베이스 트리거를 사용하여 유저 연결 정보를 캡처할 수 있습니다. 관리자는 로그인 트리거를 사용하여 데이터베이스에 연결 중인 유저를 식별하는 데이터를 캡처할 수 있습니다. 다음과 같은 예가 포함됩니다.

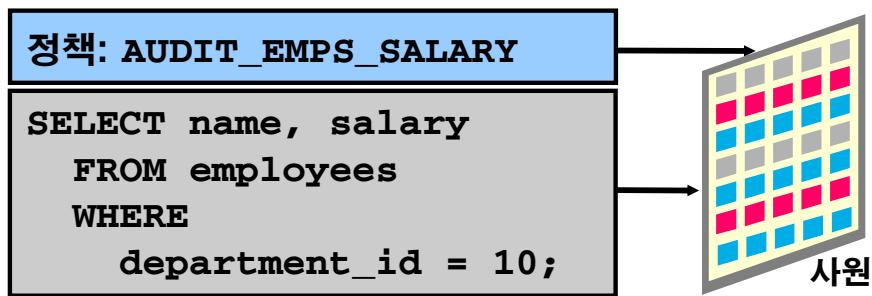
- 로그인한 사람의 IP 주소
- Instance에 연결하는 데 사용된 프로그램 이름의 첫 48자
- Instance에 연결하는 데 사용된 터미널 이름

전체 유저 파라미터 리스트는 *Oracle Database SQL Reference*의 "SYS\_CONTEXT" 섹션을 참조하십시오.

대부분의 경우 FGA(Fine-Grained Auditing) 기능이 값 기반 트리거보다 먼저 적용됩니다.

# Fine-Grained Auditing

- 컨텐트를 기준으로 데이터 액세스를 모니터
- SELECT, INSERT, UPDATE, DELETE, 및 MERGE 감사(Audit)
- 테이블이나 뷰에 있는 하나 이상의 열에 연결 가능
- 프로시저 실행 가능
- DBMS\_FGA 패키지를 사용하여 관리



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Fine-Grained Auditing

데이터베이스 감사(Audit)는 작업이 발생했다는 사실은 기록하지만 이 작업을 발생시킨 명령문에 대한 정보는 캡처하지 않습니다. FGA(Fine-Grained Auditing)는 이러한 기능을 확장한 것으로 데이터를 query 또는 조작하는 실제 SQL 문을 캡처할 수 있습니다.

또한 FGA는 표준 또는 표기 기준 데이터베이스 감사(Audit)보다 좁은 범위에서 감사(Audit)를 수행하도록 제한할 수 있습니다.

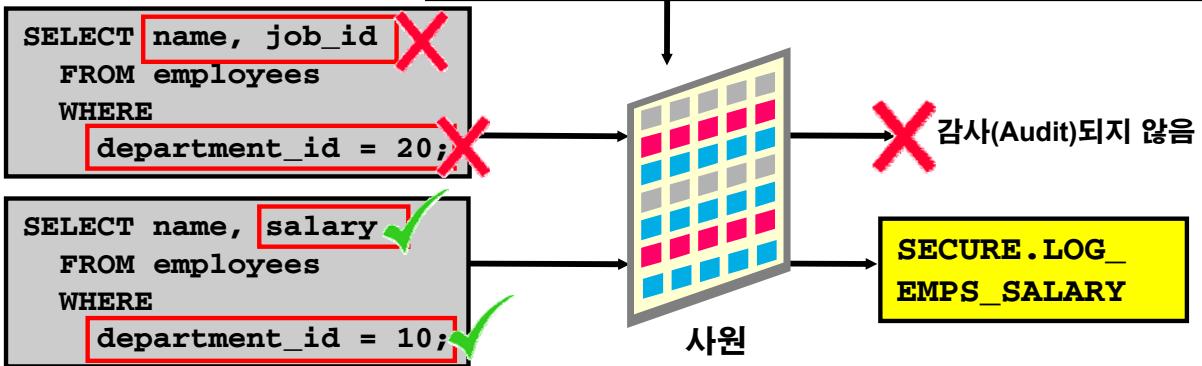
FGA 옵션을 사용하면 테이블 또는 뷰 내에 있는 개별 열을 중점적으로 감사(Audit)할 수 있으며 특정 관리자가 정의한 사양을 만족시키는 경우에만 감사(Audit)를 캡처하도록 조건을 지정할 수도 있습니다. 두 개 이상의 관련 열이 FGA 정책에 지원됩니다. 기본적으로, SQL 문에 표시된 열은 감사(Audit)됩니다. 관련된 열 중에서 임의의 열 또는 모든 열이 명령문에서 사용되는지 여부에 따라 감사(Audit)하도록 DBMS\_FGA.ALL\_COLUMNS 및 DBMS\_FGA.ANY\_COLUMNS가 제공됩니다.

DBMS\_FGA PL/SQL 패키지를 사용하여 대상 테이블 또는 뷰에 감사(Audit) 정책을 생성합니다. query 블록에서 반환된 행이 감사(Audit)된 열 및 지정된 감사(Audit) 조건과 일치하면 감사(Audit) 이벤트는 감사(Audit) 레코드를 생성하고 이를 Audit Trail에 저장합니다. 선택적으로 감사(Audit) 이벤트로 프로시저를 실행할 수도 있습니다. FGA는 자동으로 감사(Audit)를 명령문 레벨로 제한합니다. 그러므로 수천 개의 행을 반환하는 SELECT 문의 경우에도 감사(Audit) 레코드는 하나만 생성합니다.

# FGA 정책

- 정의 내용
  - 감사(Audit) 기준
  - 감사(Audit) 작업
- DBMS\_FGA.  
ADD\_POLICY를 사용하여 생성됨

```
dbms_fga.add_policy (
    object_schema  => 'HR',
    object_name    => 'EMPLOYEES',
    policy_name   => 'audit_emps_salary',
    audit_condition=> 'department_id=10',
    audit_column   => 'SALARY,COMMISSION_PCT',
    handler_schema => 'secure',
    handler_module  => 'log_emps_salary',
    enable          => TRUE,
    statement_types => 'SELECT,UPDATE');
```



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## FGA 정책

슬라이드의 예제에서는 DBMS\_FGA.ADD\_POLICY 프로시저를 사용하여 Fine-Grained Auditing 정책을 생성하는 방법을 보여줍니다. 이 프로시저는 다음 인수를 받아들입니다.

### 정책 이름

FGA 정책을 생성하는 경우 정책마다 이름을 지정합니다. 슬라이드의 예제에서는 다음 인수를 사용하여 AUDIT\_EMPS\_SALARY 정책의 이름을 지정합니다.

```
policy_name => 'audit_emps_salary'
```

### 감사(Audit) 조건

감사(Audit) 조건은 감사(Audit) 이벤트의 실행 시기를 정의하는 SQL 문입니다. 슬라이드의 예제에서는 다음 조건 인수를 사용하여 department 10의 모든 행을 감사(Audit)합니다.

```
audit_condition => 'department_id = 10'
```

주: Fine-grained audit은 Query 결과 집합을 조사하므로, 슬라이드에 표시된 FGA 정책에서는 정책 사양과 일치하는 행을 반환하는 query로 인해 감사(Audit) 레코드가 생성됩니다. 예를 들어, "select \* from employees" query에서 department\_id에 "10"이 들어 있는 모든 행이 반환될 수 있으므로, 감사(Audit) 행이 생성됩니다.

## FGA 정책(계속)

### 감사(Audit) 열

감사(Audit) 열은 감사(Audit)할 데이터를 정의합니다. 감사(Audit) 이벤트는 이 열이 SELECT 문에 포함되거나 감사(Audit) 조건을 선택할 수 있는 경우에 발생합니다. 슬라이드의 예제는 다음 인수를 사용하여 두 개의 열을 감사(Audit)합니다.

```
audit_column => 'SALARY,COMMISSION_PCT'
```

이 인수는 선택적입니다. 이 인수를 지정하지 않으면 AUDIT\_CONDITION 인수로만 감사(Audit) 이벤트 발생 여부를 결정합니다.

### 객체

객체는 감사(Audit)될 테이블 또는 뷰이며 다음의 두 인수로 전달됩니다.

- 객체를 포함하는 스키마
- 객체의 이름

슬라이드의 예제는 다음 인수를 사용하여 hr.employees 테이블을 감사(Audit)합니다.

```
object_schema => 'hr'  
object_name => 'employees'
```

### 처리기

선택적 이벤트 처리기는 감사(Audit) 중 추가로 수행해야 하는 작업을 정의하는 PL/SQL 프로시저입니다. 예를 들어, 이벤트 처리기는 관리자에게 Alert 페이지를 전송할 수 있습니다. 이벤트 처리기가 정의되지 않은 경우에는 감사(Audit) 이벤트 항목이 Audit Trail에 삽입됩니다. 감사(Audit) 이벤트 처리기가 정의된 경우에는 감사(Audit) 항목이 Audit Trail에 삽입되고 감사(Audit) 이벤트 처리기가 실행됩니다.

감사(Audit) 이벤트 항목에는 이벤트를 발생시킨 FGA 정책, SQL 문을 실행하는 유저, SQL 문 및 해당 바인드 변수가 있습니다.

이벤트 처리기는 다음 두 인수로 전달됩니다.

- PL/SQL 프로그램 단위를 포함하는 스키마
- PL/SQL 프로그램 단위의 이름

슬라이드의 예제에서는 다음 인수를 사용하여 SECURE.LOG\_EMPS\_SALARY 프로시저를 실행합니다.

```
handler_schema => 'secure'  
handler_module => 'log_emps_salary'
```

기본적으로 Audit Trail은 항상 LOB의 SQL 텍스트와 SQL 바인드 정보를 기록합니다. 예를 들어, 시스템의 성능이 저하되는 등의 경우에는 기본값이 변경될 수 있습니다.

### 상태

상태는 FGA 정책의 활성화 여부를 표시합니다. 슬라이드의 예제에서는 다음 인수로 정책을 활성화합니다.

```
enable => TRUE
```

## 감사(Audit)된 DML 문: 고려 사항

- FGA 술어가 충족되고 관련 열이 참조되는 경우 레코드를 감사(Audit)합니다.
- DELETE 문은 지정된 열에 관계없이 감사(Audit)됩니다.
- MERGE 문은 기본 INSERT, UPDATE 및 DELETE에서 생성된 문으로 감사(Audit)됩니다.

관련된 레코드 중 department 10에 대한 레코드가 없으므로 감사(Audit)되지 않습니다.

```
UPDATE hr.employees
SET salary = 1000
WHERE commission_pct = .2;
```

```
UPDATE hr.employees
SET salary = 1000
WHERE employee_id = 200;
```

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 감사(Audit)된 DML 문: 고려 사항

조작 중인 데이터 행(새 행과 기존 행 모두)이 정책 술어 조건을 충족하는 경우 DML 문에 대해 정의된 FGA 정책을 사용하여 DML 문이 감사(Audit)됩니다.

그러나 관련 열이 정책 정의에도 지정되어 있으면 데이터가 FGA 정책 술어를 충족하고 명령문이 정의된 관련 열을 참조하는 경우에 명령문이 감사(Audit)됩니다.

DELETE 문의 경우 테이블의 모든 열이 DELETE 문에 의해 처리되기 때문에 정책 정의 시 관련열을 지정하는 것은 바람직하지 않습니다. 따라서 DELETE 문은 관련 열에 관계없이 항상 감사(Audit)됩니다.

MERGE 문은 FGA에 의해 지원됩니다. 기본 INSERT, UPDATE 및 DELETE 문은 정의된 INSERT, UPDATE 또는 DELETE FGA 정책을 충족하는 경우에 감사(Audit)됩니다.

이전에 정의된 FGA 정책을 사용하는 경우 첫번째 명령문은 감사(Audit)되지 않지만 두번째 명령문은 감사(Audit)됩니다. department 10의 사원은 커미션을 받지 않지만 employee\_id=200 은 department 10의 사원을 지정합니다.

## FGA 지침

- 모든 행을 감사(Audit)하려면 null 감사(Audit) 조건을 사용합니다.
- 모든 열을 감사(Audit)하려면 null 감사(Audit) 열을 사용합니다.
- 정책 이름은 고유해야 합니다.
- 정책 생성 시에는 감사(Audit)되는 테이블 또는 뷰가 존재해야 합니다.
- 감사(Audit) 조건 구문이 유효하지 않은 경우 감사(Audit)되는 객체에 액세스하면 ORA-28112 오류가 발생합니다.
- 테이블에 감사(Audit)된 열이 없으면 어떠한 행도 감사(Audit)되지 않습니다.
- 이벤트 처리기가 없는 경우 오류가 반환되지 않고 감사(Audit) 레코드가 생성됩니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### FGA 지침

SELECT 문의 경우 FGA는 실제 행이 아니라 명령문 자체를 캡처합니다. 그러나 FGA가 Flashback Query와 결합된 경우 해당 시점에서 행이 존재하기 때문에 행이 재구성될 수 있습니다. Flashback Query에 대한 자세한 사항은 "Flashback 수행" 단원을 참조하십시오. DBMS\_FGA 패키지에 대한 자세한 사항은 *Oracle Database PL/SQL Packages and Types Reference*를 참조하십시오.

## **SYSDBA 감사(Audit)**

**SYSDBA 또는 SYSOPER 권한을 가진 유저는 데이터베이스가 닫혀 있을 때 연결할 수 있습니다.**

- Audit Trail은 데이터베이스 외부에 저장해야 합니다.
- SYSDBA 또는 SYSOPER로 연결하면 항상 감사(Audit)됩니다.
- AUDIT\_SYS\_OPERATIONS를 사용하여 SYSDBA 또는 SYSOPER 작업의 추가 감사(Audit)를 활성화할 수 있습니다.
- AUDIT\_FILE\_DEST를 사용하여 Audit Trail을 제어할 수 있습니다.



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### **SYSDBA 감사(Audit)**

SYSDBA 및 SYSOPER 유저는 데이터베이스 시작 및 종료 권한을 갖고 있습니다. 이 두 유저는 데이터베이스가 닫혀 있는 동안 데이터베이스의 내용을 변경할 수 있으므로 이 권한에 대한 Audit Trail은 데이터베이스 외부에 저장해야 합니다. 오라클 데이터베이스는 SYSDBA 및 SYSOPER 유저에 의한 로그인 이벤트를 자동으로 캡처합니다. 이 방법은 권한 여부에 관계없이 SYSDBA 및 SYSOPER 작업을 추적할 수 있지만 OS Audit Trail을 검토하는 경우에만 유용합니다.

오라클 데이터베이스는 항상 권한이 있는 유저의 로그인 이벤트를 캡처합니다. 다른 작업은 DBA 감사(Audit)를 명시적으로 활성화한 경우에만 캡처됩니다. SYSDBA 및 SYSOPER 유저에 대한 감사(Audit)는 다음 초기화 파라미터를 설정하여 활성화할 수 있습니다.

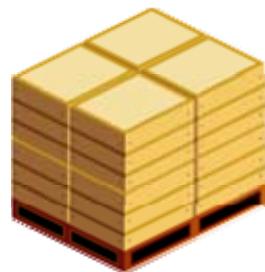
`audit_sys_operations=TRUE` (기본값: FALSE)

SYS 작업이 감사(Audit)되는 경우 AUDIT\_FILE\_DEST 초기화 파라미터는 감사(Audit) 레코드의 저장 영역 위치를 제어합니다. Windows 플랫폼에서 Audit Trail은 기본적으로 Windows 이벤트 로그에 저장되며 UNIX 및 Linux 플랫폼에서는 감사(Audit) 레코드가 AUDIT\_FILE\_DEST 위치에 저장됩니다.

# Audit Trail 유지 관리

**Audit Trail은 다음과 같은 최적의 사용법  
지침에 따라 유지 관리해야 합니다.**

- 오래된 레코드를 검토하고 저장합니다.
- 저장 영역 문제를 방지합니다.
- 레코드 손실을 피합니다.



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## Audit Trail 유지 관리

각 Audit Trail 유형은 유지 관리되어야 합니다. 기본 유지 관리에는 데이터베이스 또는 운영 체제에서 감사(Audit) 레코드를 검토하고 오래된 레코드를 제거하는 작업이 포함됩니다. Audit Trail을 늘려 사용 가능한 저장 영역을 채울 수 있습니다. 파일 시스템이 가득 차면 시스템 고장 또는 단순한 성능 문제가 발생할 수 있습니다. 데이터베이스 Audit Trail이 테이블스페이스를 채운 경우 감사(Audit)된 작업이 완료되지 않습니다. Audit Trail이 시스템 테이블스페이스를 채운 경우 감사(Audit) 작업이 종료되기 전에 다른 작업의 성능에 영향을 줍니다.

표준 감사(Audit)의 Audit Trail은 AUD\$ 테이블에 저장됩니다. FGA의 Audit Trail은 FGA\_LOG\$ 테이블에 저장됩니다. 두 테이블 모두 기본적으로 SYSTEM 테이블스페이스에 생성됩니다. Data Pump 엑스포트/임포트 유틸리티를 사용하여 이러한 테이블을 다른 테이블스페이스로 이동할 수 있습니다.

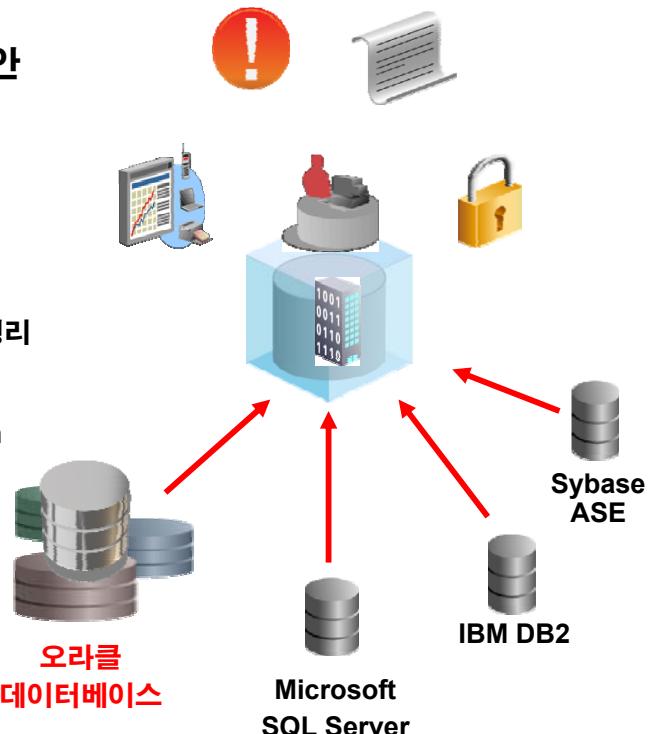
**주:** 감사(Audit) 테이블을 SYSTEM 테이블스페이스 밖으로 이동하는 것은 지원되지 않습니다. 감사(Audit) 테이블에서 레코드를 제거하는 과정에 감사(Audit) 레코드를 잃을 수도 있습니다.

### 최적의 사용법(Best Practice Tip)

시간 기록에 따라 엑스포트를 사용한 다음 동일한 시간 기록에 따라 Audit Trail에서 행을 삭제하십시오.

# Oracle Audit Vault

- **감사(Audit) 데이터 통합 및 보안**
  - Oracle 9i Release 2 이상
  - SQL Server 2000, 2005
  - IBM DB2 UDB 8.5 & 9.2
  - Sybase ASE 12.5 - 15.0
  - 보안 및 확장
  - 소스 오라클 감사(Audit) 데이터 정리
- **중앙화된 보고**
  - 널리 사용되는 Oracle Application Express를 사용한 간편한 보고서 인터페이스
  - 준수를 위한 표준 보고서
  - 새 커스텀 보고서
- **보안 위협에 대한 Alert**
  - 보안 관련 이벤트에 대한 검색 및 Alert



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Oracle Audit Vault

Oracle Audit Vault의 주요 이점은 다음과 같습니다.

- Oracle Audit Vault는 Oracle9i Database Release 2부터 시작되는 오라클 데이터베이스, Microsoft SQL Server 2000 & 2005, IBM DB2 Unix, Linux, Windows 8.2 & 9.5 및 Sybase ASE 12.5 – 15.0에서 감사(Audit) 데이터를 투명하게 수집하고 통합합니다.
- Oracle Audit Vault는 조직에서 내장 보고서 및 커스텀 보고서를 통해 준수 보고를 간소화할 수 있습니다. 또한 Oracle Audit Vault에서는 Oracle BI Publisher, Oracle Application Express 또는 Third-Party 보고 도구에서 액세스할 수 있는 개방형 감사(Audit) 웨어하우스 스키마를 제공합니다.
- Oracle Audit Vault는 의심스러운 활동에 대해 alert를 제공하므로 내부자의 위협을 감지 및 방지할 수 있습니다.
- Oracle Audit Vault는 오라클의 데이터 웨어하우징 기술을 기반으로 구축되었으며, Oracle Database Vault 및 Oracle Advanced Security 등의 오라클 데이터베이스 보안 제품을 통해 보호되는 안전하고 확장성 있는 감사(Audit) 웨어하우스가 포함됩니다. Oracle Audit Vault에는 관리 효율성과 성능을 개선하기 위한 Oracle Partitioning이 포함되어 있습니다.
- Oracle Audit Vault는 중앙화된 데이터베이스 감사(Audit) 설정(정책) 관리를 통해 IT 비용을 낮추고 IT 보안 관리자 및 내부 감사자가 작업을 보다 쉽게 수행할 수 있게 해줍니다.

자세한 내용은 Audit Vault 설명서를 참조하십시오.

## 퀴즈

표준 데이터베이스 감사(Audit)는 DML 트랜잭션의 사전 및 사후 변경 사항을 캡처합니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 퀴즈

**SYSDBA 및 SYSOPER 작업에 대한 감사(Audit)는 기본적으로 활성화됩니다.**

- 1. 참**
- 2. 거짓**



Copyright © 2010, Oracle. All rights reserved.

**해답: 2**

# 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 보안 및 감사(Audit)를 위한 DBA 책임 설명
- 표준 데이터베이스 감사(Audit) 활성화
- 감사(Audit) 옵션 지정
- 감사(Audit) 정보 검토
- Audit Trail 유지 관리



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 11 개요: 오라클 데이터베이스 보안 구현

이 연습에서는 다음 내용을 다룹니다.

- 표준 데이터베이스 감사(Audit) 활성화
- HR.JOBES 테이블에 대해 감사(Audit) 옵션 지정
- 테이블 갱신
- 감사(Audit) 정보 검토
- Audit Trail 유지 관리

ORACLE®

Copyright © 2010, Oracle. All rights reserved.



# 데이터베이스 유지 관리

# 12

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

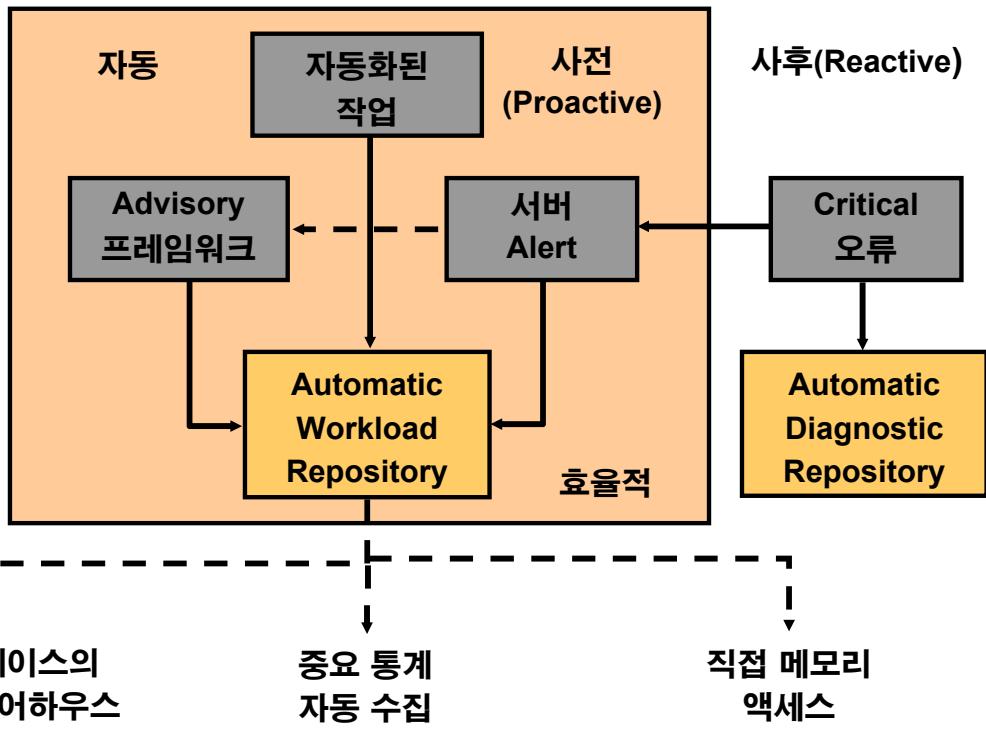
이 단원을 마치면 다음을 수행할 수 있습니다.

- 옵티마이저 통계 관리
- AWR(Automatic Workload Repository) 관리
- ADDM(Automatic Database Diagnostic Monitor) 사용
- advisory 프레임워크 설명 및 사용
- alert 임계값 설정
- Server-Generated Alert 사용
- 자동화된 업무 사용

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

# 데이터베이스 유지 관리



Copyright © 2010, Oracle. All rights reserved.

## 데이터베이스 유지 관리

다음과 같은 주요 요소를 포함하는 오라클 데이터베이스의 정교한 Infrastructure를 사용하면 사전(Proactive) 데이터베이스 유지 관리를 손쉽게 수행할 수 있습니다.

- AWR(Automatic Workload Repository)은 각 오라클 데이터베이스에 내장된 Repository입니다. 오라클 데이터베이스 서버는 정기적으로 모든 주요 통계 및 작업 로드 정보의 스냅샷을 만들고 이 데이터를 AWR에 저장합니다. 관리자나 데이터베이스 자체(또는 두 가지 모두)에서 캡처된 데이터를 분석할 수 있습니다.
- 데이터베이스는 자동화된 업무를 사용하여 정기적 백업, 옵티마이저 통계 Refresh, 데이터베이스 상태 검사 등과 같은 일상적인 유지 관리 작업을 수행합니다.

사후(Reactive) 데이터베이스 유지 관리 작업에는 데이터베이스 상태 검사기를 통해 발견되는 Critical 오류 및 조건이 포함됩니다.

- 자동으로 해결할 수 없고 관리자에게 알려야 하는 문제(예: 공간 부족)에 대해 오라클 데이터베이스 서버는 Server-generated alert를 제공합니다. 오라클 데이터베이스 서버는 기본적으로 자체 모니터되며 문제를 통지하기 위한 Alert를 보냅니다. 이 Alert는 관리자에게 통지되며 보고된 문제를 해결하기 위한 권장 사항을 제공하기도 합니다.
- 권장 사항은 각각 서브 시스템을 담당하는 수많은 Advisor로부터 생성됩니다. 예를 들어, Memory Advisor, Segment Advisor, SQL Advisor 등이 있습니다.

# Alert History 확인

**Related Links**

<a href="#">Access</a>	<a href="#">Add Exadata Cell Targets</a>	<a href="#">Advisor Central</a>
<a href="#">Alert History</a>	<a href="#">Alert Log Contents</a>	<a href="#">All Metrics</a>
<a href="#">Baseline Metric Thresholds</a>	<a href="#">Blackouts</a>	<a href="#">EM SQL History</a>
<a href="#">Jobs</a>	<a href="#">Metric and Policy Settings</a>	<a href="#">Metric Collection Errors</a>
<a href="#">Monitoring Configuration</a>	<a href="#">Monitor in Memory Access Mode</a>	<a href="#">Policy Groups</a>
<a href="#">Scheduler Central</a>	<a href="#">SQL Worksheet</a>	<a href="#">Target Properties</a>
<a href="#">User-Defined Metrics</a>		

**Alert History**

Page Refreshed Jun 19, 2009 11:40:17 PM GMT+07:00  
View Data

Metric	History	Key
Audited User		Critical Warning Clear No Data
Instance Status		
Mounted		

11:40 3 6 9 12 PM 3 6 9  
18 June 2009

Severity	Timestamp ▾	Message	Last Comment	Details
✓	Jun 19, 2009 2:02:12 AM	The instance is down, and health check reported: ..		-
✗	Jun 19, 2009 1:32:12 AM	The instance is down, and health check reported: Instance Shutdown.		-
✓	Jun 18, 2009 5:31:15 AM	The instance is down, and health check reported: ..		-

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## Alert History 확인

Alert History 페이지는 현재 데이터베이스의 Alert 기록을 보여주는 차트를 지정된 시간 단위로 표시합니다. Alert는 모니터된 metric의 Warning 또는 Critical 임계값이나 대상을 더 이상 사용할 수 없는 경우 등의 잠재적인 문제를 나타냅니다. 각 Alert에 대한 자세한 통계, 그래프 및 실제 시간 기록을 얻으려면 Alert History 페이지에 나열된 Metric 이름을 누르십시오. 또한 해결 정보와 같은 Alert 관련 설명을 입력할 위치도 있습니다.

## 용어

- **AWR(Automatic Workload Repository): 데이터 수집, 분석 및 권장 해결 방안을 위한 Infrastructure**
- **AWR Baseline: 성능 비교를 위한 AWR 스냅샷 집합**
- **Metric: 누적 통계의 변화율**
- **통계: 데이터베이스 및 객체 세부 정보를 제공하는 데이터 모음**
  - 옵티마이저 통계: query 옵티마이저에서 사용됨
  - 데이터베이스 통계: 성능을 위해 사용됨
- **임계값: metric 값이 비교되는 경계 값**



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 용어

AWR(Automatic Workload Repository)은 내부 Oracle 서버 구성 요소가 문제 감지 및 자체 튜닝 목적을 위해 성능 통계를 수집, 처리, 유지 관리 및 활용할 수 있는 서비스를 제공합니다.

ASH(Active Session History)는 AWR에 저장된 최근 세션 작업의 기록입니다.

통계는 데이터베이스와 데이터베이스의 객체에 대한 세부 정보를 제공하는 데이터 모음입니다. Query 옵티마이저는 옵티마이저 통계를 사용하여 각 SQL 문에 가장 적합한 실행 계획을 선택합니다. 데이터베이스 통계는 성능 모니터를 위한 정보를 제공합니다.

AWR 스냅샷에는 데이터베이스 통계 및 Metrics, 응용 프로그램 통계(트랜잭션 볼륨, 응답 시간), 운영 체제 통계 및 기타 측정 값이 포함됩니다. AWR baseline은 시간의 흐름에 따라 수집한 AWR 스냅샷 집합으로, 현재 성능과 Baseline 비교 또는 Baseline 간의 비교와 같이 성능을 비교할 때 사용됩니다.

Oracle Database 11g에서는 기본적으로 System Moving Window Baseline이 수집됩니다.

System Moving Window Baseline은 지난 8일 동안의 스냅샷이 기본적으로 포함된 변경하는 스냅샷 집합입니다. 이 Baseline은 충분한 데이터가 수집되고 통계가 계산되면 유효한 상태가 됩니다. 기본적으로 통계 계산은 매주 토요일 자정으로 일정이 잡힙니다.

## Oracle 옵티마이저: 개요

**Oracle 옵티마이저는 가장 효율적인 실행 계획을 결정하며 SQL 문 처리에서 가장 중요한 단계입니다.**

### 옵티마이저:

- 표현식 및 조건을 평가합니다.
- 객체 및 시스템 통계를 사용합니다.
- 데이터 액세스 방법을 결정합니다.
- 테이블 조인 방법을 결정합니다.
- 가장 효율적인 경로를 결정합니다.

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### Oracle 옵티마이저: 개요

옵티마이저는 SQL 문에 대한 실행 계획을 생성하는 오라클 데이터베이스의 일부입니다. 실행 계획의 결정은 SQL 문을 처리하는 데 있어서 중요한 단계이며, 실행 시간에 큰 영향을 줄 수 있습니다.

실행 계획이란 명령문을 실행하기 위해 순서대로 수행되는 일련의 작업을 말합니다. 옵티마이저는 query에서 지정된 조건과 참조되는 객체와 관련된 많은 요소들을 고려합니다. 옵티마이저에 필요한 정보는 다음과 같습니다.

- 시스템(I/O, CPU 등)뿐만 아니라 스키마 객체(행 수, 인덱스 등)에 대해 수집된 통계
- 덕셔너리의 정보
- WHERE 절 수식자
- 개발자가 제공한 힌트

Enterprise Manager, EXPLAIN PLAN 및 SQL\*Plus AUTOTRACE와 같은 진단 도구를 사용할 때는 옵티마이저가 선택하는 실행 계획을 볼 수 있습니다.

**주:** Oracle 옵티마이저는 기능에 따라 *query 옵티마이저* 및 *자동 튜닝 옵티마이저*라는 두 개의 이름을 가집니다.

# 옵티마이저 통계

옵티마이저 통계는 다음 특징을 갖습니다.

- 특정 시점의 스냅샷
- instance 재시작 시에도 영구적으로 유지됨
- 자동으로 수집됨

```
SQL> SELECT COUNT(*) FROM hr.employees;
      COUNT(*)
-----
      214
SQL> SELECT num_rows FROM dba_tables
  2 WHERE owner='HR' AND table_name = 'EMPLOYEES';
      NUM_ROWS
-----
      107
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 옵티마이저 통계

옵티마이저 통계에는 테이블, 열, 인덱스 및 시스템 통계가 포함됩니다. 테이블 및 인덱스에 대한 통계는 데이터 딕셔너리에 저장됩니다. 이러한 통계는 실시간 데이터를 제공하기 위한 것이 아니라 옵티마이저에 데이터 저장과 분산에 관해 통계적으로 올바른 스냅샷을 제공하기 위한 것으로, 이를 사용하여 옵티마이저는 데이터 액세스 방법을 결정할 수 있습니다.

수집되는 통계에는 다음이 포함됩니다.

- 데이터베이스 블록에 있는 테이블 또는 인덱스의 크기
- 행 수
- 평균 행 크기 및 체인 수(테이블에만 해당)
- 삭제된 최하위 행의 높이 및 수(인덱스에만 해당)

데이터가 삽입, 삭제 및 수정되면 이러한 값이 변경됩니다. 데이터 분산 통계를 실시간으로 유지 관리하면 성능에 상당한 영향을 미치게 되므로 이러한 통계는 테이블 및 인덱스에서 통계를 주기적으로 수집하여 갱신합니다.

옵티마이저 통계는 기본적으로 하루에 한 번씩 미리 정의된 유지 관리 윈도우(Maintenance windows) 동안 실행되는 자동 유지 관리 작업에 의해 자동으로 수집됩니다. 시스템 통계는 옵티마이저가 사용하는 운영 체제 특성입니다. 이러한 통계는 자동으로 수집되지 않습니다. 시스템 통계 수집에 대한 자세한 사항은 *Oracle Database Performance Tuning Guide*를 참조하십시오.

옵티마이저 통계는 AWR 스냅샷에서 수집되는 데이터베이스 성능 통계와는 다릅니다.

# Manage Optimizer Statistics 페이지 사용

The screenshot shows the Oracle Enterprise Manager interface for a database instance named `orcl.oracle.com`. The top navigation bar includes links for Home, Performance, Availability, Server (which is selected), Schema, Data Movement, and Software and Support.

In the left sidebar under the **Query Optimizer** section, there are three links: **Manage Optimizer Statistics**, **Gather Optimizer Statistics**, and **SQL Plan Control**. The **Manage Optimizer Statistics** link is highlighted with a red box and a red arrow points to it from the left.

The main content area is titled **Manage Optimizer Statistics** and displays the following information:

- Database:** `orcl.oracle.com`
- Description:** Optimizer Statistics are used by the query optimizer to choose the best execution plan for each SQL statement. Up-to-date optimizer statistics can greatly improve the performance of SQL statements.
- Operations:**
  - [Gather Optimizer Statistics](#)
  - [Restore Optimizer Statistics](#)
  - [Lock Optimizer Statistics](#)
  - [Unlock Optimizer Statistics](#)
  - [Delete Optimizer Statistics](#)
- Related Links:**
  - [Object Statistics](#)
  - [Global Statistics Gathering Options](#)
  - [Object Level Statistics Gathering Preferences](#)
  - [Job Scheduler](#)
  - [Automated Maintenance Tasks](#)

At the bottom right of the interface is the **ORACLE** logo, and at the bottom center is the copyright notice: Copyright © 2010, Oracle. All rights reserved.

## Manage Optimizer Statistics 페이지 사용

Enterprise Manager에서 옵티마이저 통계를 관리하려면 Server 탭을 누른 다음 Query Optimizer 섹션 아래에 있는 Manage Optimizer Statistics를 누릅니다. 이 페이지에서 다음과 같은 통계 업무를 수행할 수 있습니다.

- 수동으로 옵티마이저 통계를 수집합니다.
- 옵티마이저 통계를 과거의 특정 시점으로 복원합니다. 선택한 시점은 옵티마이저 통계 Retention 기간 이내여야 하며 기본값은 30일입니다.
- 특정 객체에 대한 통계를 겹쳐쓰지 않도록 옵티마이저 통계를 Lock합니다. 잘 표현된 데이터가 존재하는 시간에 특정 테이블에 대해 통계가 계산되었을 때 이러한 통계를 항상 보유하려는 경우 유용합니다. 테이블이 Lock된 경우 테이블의 변동은 통계에 영향을 주지 않습니다.
- 이전에 수행된 Lock을 언두하려면 옵티마이저 통계의 Lock을 해제합니다.
- 통계를 삭제하려면 옵티마이저 통계를 삭제합니다.

### 최적의 사용법(Best Practice Tip)

옵티마이저 통계를 수집하려면 자동 유지 관리 업무를 사용합니다. 옵티마이저 통계를 수집하도록 업무를 설정하려면 STATISTICS\_LEVEL 초기화 파라미터를 TYPICAL 또는 ALL로 설정해야 합니다.

# 수동으로 옵티마이저 통계 수집

**Manage Optimizer Statistics**

Database: orcl.oracle.com

Optimizer Statistics are used by the query statistics can greatly improve the performance.

**Operations**

- Gather Optimizer Statistics** (highlighted with a red box)
- Restore Optimizer Statistics
- Lock Optimizer Statistics
- Unlock Optimizer Statistics
- Delete Optimizer Statistics

**Information**

For 11, Oracle recommends you enable automated maintenance task (Optimizer Statistics Gathering) to generate optimizer statistics regularly within maintenance windows. This wizard should only be used for cases where the task is inappropriate or disabled. For example, you may want to gather optimizer statistics immediately, or the task failed to execute within a maintenance window, or you want to customize options to gather optimizer statistics.

**Gather Optimizer Statistics: Scope**

Database: orcl.oracle.com      Logged In As: SYS      Cancel      Step 1 of 5      Next

Task Status: Enabled      Scope: Database

Select the type of object for which you want to gather optimizer statistics.

Object Type:

- Database
- Schema
- Tables
- Indexes
- Fixed Objects
- Dictionary Objects

In-memory structures/variables of the RDBMS that are exposed in the form of dynamic performance tables.

Objects in 'SYS', 'SYSTEM' and all non-user defined schemas.

**TIP** The Objects step will be skipped when Database, Fixed Objects or Dictionary Objects is selected.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 수동으로 옵티마이저 통계 수집

자동 수집 작업 간에 테이블 내용이 너무 많이 변경되어 통계가 더 이상 테이블을 정확하게 나타내지 않는 경우 등의 특정 상황에서는 통계를 수동으로 수집해야 할 수 있습니다.

24시간 동안 크기가 10% 이상 변경되는 대형 테이블의 경우 이러한 현상이 자주 발생합니다.

**최적의 사용법(Best Practice Tip):** 수집 기간 사이에 테이블이 10% 이상 변경되지 않도록 통계를 자주 수집하십시오. 이렇게 하려면 통계를 수동으로 수집하거나 유지 관리 윈도우(Maintenance windows)가 추가로 필요할 수 있습니다.

Enterprise Manager 또는 DBMS\_STATS 패키지를 사용하면 통계를 수동으로 수집할 수 있습니다. 시스템 통계는 DBMS\_STATS 패키지를 통해서만 수집할 수 있습니다. 시스템 통계는 I/O 및 CPU 성능과 활용률 같은 하드웨어 특성을 Query 옵티마이저에 설명합니다.

Gather Optimizer Statistics 메뉴를 선택하면 옵티마이저 통계를 수집할 작업의 범위, 객체, 옵션 및 일정을 선택할 수 있는 마법사가 시작됩니다. 이 마법사는 유저가 지정하는 범위(테이블, 스키마 또는 데이터베이스)에서 DBMS\_STATS.GATHER\_\*\_STATS 작업을 제출(Submit)합니다. 이 마법사에서는 DBMS\_STATS 패키지가 사용하는 기본값에 대한 환경 설정을 지정하고 원하는 시간에 이 작업이 실행되도록 일정을 잡습니다.

## 수동으로 옵티마이저 통계 수집(계속)

통계는 유지 관리 윈도우(Maintenance windows) 동안에 보다 효율적으로 수집되며 유저에게 미치는 영향도 적으므로 일상적인 통계 수집을 위해 통계를 수동으로 수집하는 작업은 권장되지 않습니다. 자동 작업이 실패하거나 비활성화된 경우에도 수동 작업을 제출(submit) 할 수 있습니다.

DBMS\_STATS 패키지를 사용하면 옵티마이저 통계를 직접 수집할 수도 있습니다.

```
SQL> EXEC dbms_stats.gather_table_stats('HR', 'EMPLOYEES');
SQL> SELECT num_rows FROM dba_tables
  2 WHERE owner='HR' AND table_name = 'EMPLOYEES';
  NUM_ROWS
-----
      214
```

행 수는 통계가 수집될 당시 테이블에 있던 행 수를 정확히 반영합니다. DBMS\_STATS는 전체 스키마 또는 전체 데이터베이스에 대해서도 수동으로 통계를 수집할 수 있습니다.

작업 로드가 크게 변경되는 경우가 아니면 시스템 통계는 변경되지 않습니다. 그러므로 시스템 통계는 자주 조정할 필요가 없습니다. DBMS\_STATS.GATHER\_SYSTEM\_STATS 프로시저는 지정된 기간 동안 시스템 통계를 수집합니다. 시스템 통계 수집을 시작한 다음 다른 호출을 통해 수집을 정지할 수도 있습니다.

**최적의 사용법(Best Practice Tip):** 데이터베이스를 생성할 때는 다음 명령을 사용하십시오.

```
SQL> EXEC dbms_stats.gather_system_stats('NOWORKLOAD');
```

NOWORKLOAD 옵션은 데이터베이스 크기에 따라 몇 분이 지난 후에 평균 읽기 탐색 시간 및 I/O 전송률 등의 I/O 특성 예상치를 캡처합니다.

# 통계 수집 환경 설정



옵티マイ저  
통계 수집 작업

## 범위

명령문 레벨

테이블 레벨

스키마 레벨

데이터베이스 레벨

GLOBAL 레벨

## 환경 설정

CASCADE

DEGREE

ESTIMATE\_PERCENT

NO\_INVALIDATE

METHOD\_OPT

GRANULARITY

INCREMENTAL

PUBLISH

STALE\_PERCENT



DBMS\_STATS

set | get | delete | export | import

```
exec dbms_stats.set_table_prefs('SH','SALES','STALE_PERCENT','13');
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 통계 수집 환경 설정

DBMS\_STATS.GATHER\_\*\_STATS 프로시저를 다양한 레벨에서 호출하여 전체 데이터베이스 또는 테이블 등의 개별 객체에 대한 통계를 수집할 수 있습니다. GATHER\_\*\_STATS 프로시저를 호출하면 다양한 파라미터가 기본값으로 허용됩니다. 제공되는 기본값은 데이터베이스에 포함된 대부분의 객체에 대해 문제 없이 작동하지만, Oracle Database 11g에서는 이러한 각 객체에 대해 수동 작업을 실행하는 대신 개별 객체, 스키마 또는 데이터베이스에 대해 환경 설정이라는 값을 설정하거나, Global 레벨 명령을 통해 기본값을 변경할 수 있습니다.

환경 설정은 수집 프로시저에 제공되는 파라미터를 지정합니다. SET\_\*\_PREFS 프로시저는 SYS 또는 SYSTEM이 소유하지 않은 모든 객체에 대해 환경 설정 값을 생성합니다. 이 프로시저는 DBA가 데이터베이스 차원에 적용되어야 하는 파라미터에 대한 Global 환경 설정을 지정할 때 사용됩니다. 이러한 프로시저는 기본값으로 허용되는 모든 파라미터에 적용됩니다.

SET\_DATABASE\_PREFS 프로시저는 지정한 환경 설정을 설정하는 데이터베이스의 모든 테이블 및 스키마를 반복합니다. SET\_SCHEMA\_PREFS는 지정된 스키마의 테이블을 반복합니다. SET\_TABLE\_PREFS는 단일 테이블에 대한 환경 설정 값을 설정합니다.

모든 객체 환경 설정(데이터베이스, 스키마 또는 테이블 레벨 등 설정 레벨은 관계없음)은 단일 테이블에 보관됩니다. 스키마 레벨에서 환경 설정을 변경하면 이전에 테이블 레벨에서 설정한 환경 설정이 겹쳐쓰여집니다.

## 통계 수집 환경 설정(계속)

다양한 수집 프로시저는 실행 시에 각 객체에 대해 설정한 객체 레벨 환경 설정을 검색합니다. DBA\_TAB\_STAT\_PREFS 뷰의 객체 레벨 환경 설정을 볼 수 있습니다. 객체 레벨에서 설정하지 않은 모든 환경 설정은 Global 레벨 환경 설정으로 설정됩니다. Global 환경 설정은 각 환경 설정에 대해 DBMS\_STATS.GET\_PREFS 프로시저를 호출하여 볼 수 있습니다.

이러한 환경 설정은 테이블, 스키마, 데이터베이스 및 Global 레벨에서 설정하고, 가져오고, 삭제하고, 엑스포트하고, 임포트할 수 있습니다. 환경 설정 값은 Global에서 테이블 레벨로 설정되며, 가장 작은 마지막 그룹에 환경 설정을 적용합니다.

Oracle Database 11g의 환경 설정:

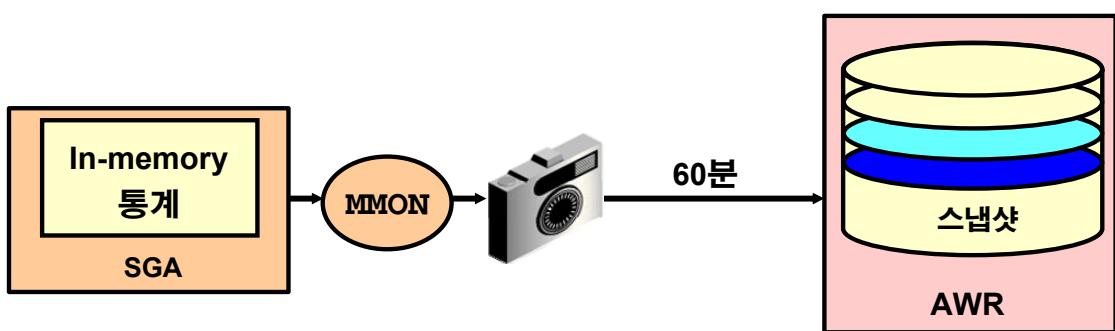
- CASCADE는 인덱스 통계가 테이블 통계 수집의 일부로 수집되는지 여부를 결정합니다.
- DEGREE는 통계 수집 시 사용되는 병렬도를 설정합니다.
- PUBLISH는 통계를 딕셔너리에 게시(Publish)할 것인지 아니면 전용(Private) 영역에 저장할 것인지를 결정하는 데 사용됩니다. 그러면 DBA가 PUBLISH\_PENDING\_STATS 프로시저를 사용하여 통계를 데이터 딕셔너리에 게시하기 전에 검증할 수 있습니다.
- STALE\_PERCENT는 객체의 통계가 오래된 것인지를 판단할 때 기준으로 삼을 임계값 레벨을 결정하는 데 사용됩니다. 이 값은 마지막 통계 수집 이후 수정된 행의 비율입니다. 예제는 SH.SALES에 대해서만 기본값 10%를 13%로 변경합니다.
- INCREMENTAL은 Partitioning된 테이블의 Global 통계를 Incremental 방식으로 수집하는 데 사용됩니다.
- METHOD\_OPT는 열 통계를 수집하는 데 사용되는 열 및 히스토그램 파라미터를 결정합니다.
- GRANULARITY는 수집 할 통계의 세분화 정도를 결정합니다(Partition 테이블의 경우에만 해당).
- NO\_INVALIDATE는 커서를 무효화할지 여부를 결정하는 데 사용됩니다.
- ESTIMATE\_PERCENT는 올바른 통계를 얻기 위해 샘플링 할 행 수를 결정하는 데 사용됩니다. 테이블에 있는 행 수의 비율입니다.

**주:** 환경 설정에 대한 자세한 사항은 *Oracle Database PL/SQL Packages and Types Reference*의 DBMS\_STATS 설명서를 참조하십시오.

테이블, 스키마 및 데이터베이스 레벨에서 DBMS\_STATS.DELETE\_\*\_PREFS 프로시저를 사용하면 환경 설정이 삭제될 수 있습니다. 그리고 DBMS\_STATS.RESET\_PARAM\_DEFAULTS 프로시저를 사용하면 Global 환경 설정을 권장 값으로 재설정할 수 있습니다.

# AWR(Automatic Workload Repository)

- 성능 정보에 대한 내장 Repository
- 60분마다 데이터베이스 Metrics의 스냅샷을 생성하여 8일간 보관
- 모든 자체 관리 기능(self-management)의 토대



ORACLE

Copyright © 2010, Oracle. All rights reserved.

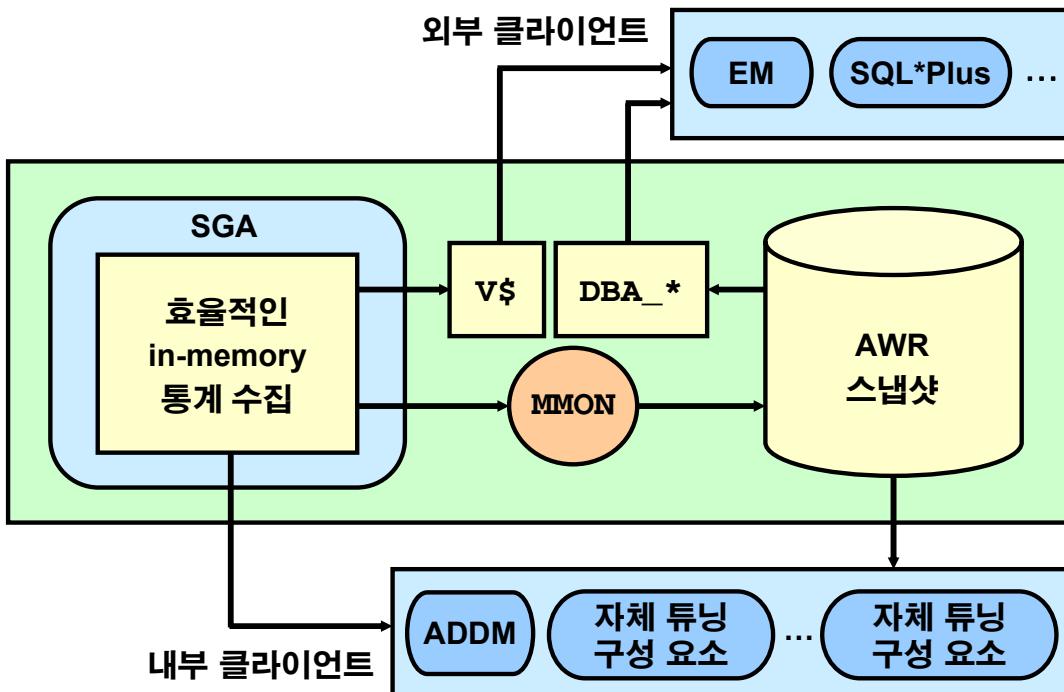
## AWR(Automatic Workload Repository)

AWR은 Oracle Database 11g 구성 요소가 문제 감지 및 자체 튜닝 목적을 위해 통계를 수집, 유지 관리 및 활용할 수 있도록 서비스를 제공하는 Infrastructure입니다. 데이터베이스 통계, Metrics 등에 대한 데이터 웨어하우스로 간주할 수 있습니다.

기본적으로 데이터베이스는 SGA에서 60분마다 자동으로 통계 정보를 캡처하여 스냅샷의 형태로 AWR에 저장합니다. 이러한 스냅샷은 MMON(Manageability Monitor)이라는 백그라운드 프로세스에 의해 디스크에 저장됩니다. 기본적으로 스냅샷은 8일간 보관됩니다. 스냅샷 간격과 Retention 간격은 모두 수정할 수 있습니다.

AWR에는 수백 개의 테이블이 포함되어 있으며 이 테이블은 모두 SYSMAN 스키마에 속해 있으며 SYSAUX 테이블스페이스에 저장되어 있습니다. 오라클에서는 Enterprise Manager 또는 AWR에 사용할 DBMS\_WORKLOAD\_REPOSITORY 패키지를 통해서만 Repository에 액세스할 것을 권장합니다. Repository 테이블에 대한 직접 DML은 지원되지 않습니다.

# AWR Infrastructure



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## AWR Infrastructure

AWR Infrastructure는 크게 다음 두 부분으로 구성됩니다.

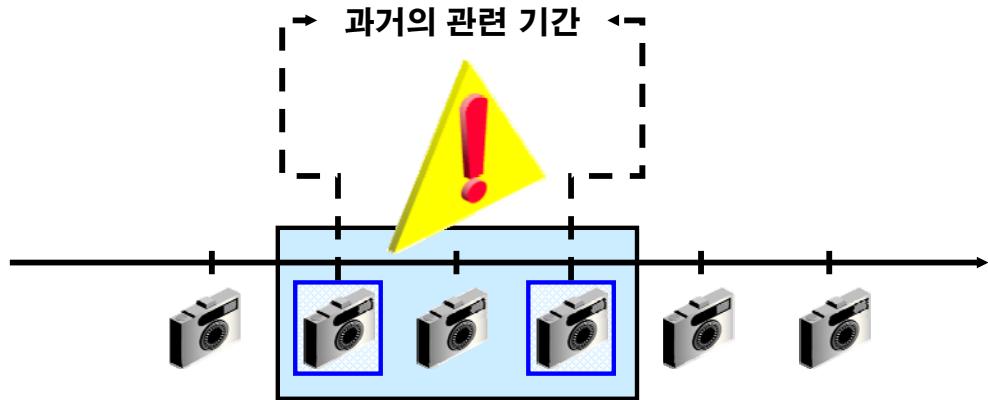
- Oracle Database 11g 구성 요소에서 통계 수집에 사용하는 In-memory 통계 수집 서비스. 이 통계는 성능상의 이유로 메모리에 저장됩니다. 메모리에 저장된 통계는 Dynamic performance (V\$) 뷰를 통해 액세스할 수 있습니다.
- 서비스의 지속 부분을 나타내는 AWR 스냅샷. AWR 스냅샷은 데이터 딕셔너리 뷰와 Enterprise Manager Database Control을 통해 액세스할 수 있습니다.

통계는 다음 이유로 인해 지속 영역에 저장됩니다.

- 통계는 Instance Crash가 발생해도 유지되어야 합니다.
- 일부 분석에는 Baseline 비교를 위해 과거 기록 데이터가 필요합니다.
- 메모리 오버플로우가 발생할 수 있습니다. 메모리 부족으로 인해 오래된 통계가 새 통계로 바뀔 때 나중에 사용하기 위해 이전 데이터를 저장할 수 있습니다.

통계의 메모리 버전은 MMON 백그라운드 프로세스에 의해 정기적으로 디스크에 전달됩니다. AWR을 통해 오라클 데이터베이스는 DBA의 개입 없이 기록 데이터를 자동으로 캡처할 수 있도록 합니다.

# AWR Baseline



```
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE ( -
    start_snap_id IN NUMBER,
    end_snap_id   IN NUMBER,
    baseline_name IN VARCHAR2 );
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## AWR Baseline

AWR baseline은 AWR 스냅샷의 집합입니다. 일반적으로 AWR baseline은 AWR에서 태그를 지정하고 보존하는 중요 기간에 대한 스냅샷 데이터의 집합입니다. Baseline은 스냅샷의 쌍으로 정의되며, 스냅샷은 스냅샷 시퀀스 번호(snap\_id) 또는 시작 및 종료 시간으로 식별됩니다. 각 스냅샷 집합에는 시작 및 종료 스냅샷이 있고 그 사이에 모든 스냅샷이 포함되어 있습니다. 스냅샷 집합은 스냅샷 데이터를 보유하는 데 사용됩니다. 따라서 기본적으로 스냅샷 집합에 포함된 스냅샷은 스냅샷 집합이 삭제될 때까지 유지됩니다. 만기 값은 스냅샷을 보유할 일수로 설정할 수 있습니다.

Baseline은 유저가 제공한 이름으로 식별됩니다. CREATE\_BASELINE 프로시저를 실행하여 스냅샷 집합에서 Baseline을 생성하고 이를 및 스냅샷 식별자의 쌍을 지정합니다. 데이터베이스가 존재하는 동안 고유한 Baseline 식별자가 새로 생성된 Baseline에 지정됩니다. 일반적으로 현재의 시스템 동작과 비교하는 데 사용할 과거의 대표적인 기간으로부터 Baseline을 설정합니다. 또한 Database Control에서 Baseline을 사용하여 임계값 기반 Alert를 설정할 수도 있습니다. 이 프로시저의 만료 파라미터를 사용하여 만료 시간을 일수로 설정할 수 있습니다. 기본값은 NULL, 즉 "만료 없음"입니다.

DBA\_HIST\_SNAPSHOT 또는 Database Control에서 직접 snap\_id를 가져올 수 있습니다.

**주:** DBMS\_WORKLOAD\_REPOSITORY 패키지에 대한 자세한 내용은 *Oracle Database PL/SQL Packages and Types Reference*를 참조하십시오.

# Enterprise Manager 및 AWR

The screenshot shows the Oracle Enterprise Manager interface with the 'Server' tab selected. On the left, under 'Statistics Management', the 'Automatic Workload Repository' link is highlighted with a red box and a red arrow pointing to it from below.

**Automatic Workload Repository**

Page Refreshed Aug 21, 2008 9:47:11 PM MDT [Refresh](#)

The Automatic Workload Repository is used for storing database statistics that are used for performance tuning.

**General**

[Edit](#)

- Snapshot Retention (days) **8**
- Snapshot Interval (minutes) **60**
- Collection Level **TYPICAL**

Next Snapshot Capture Time **Aug 21, 2008 10:00:49 PM**

**Manage Snapshots and Baselines**

[Run AWR Report](#)

- Snapshots **195**
- Baselines **1**

Latest Snapshot Time **Aug 21, 2008 9:00:49 PM**  
Earliest Snapshot Time **Aug 13, 2008 7:00:26 PM**

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

## Enterprise Manager 및 AWR

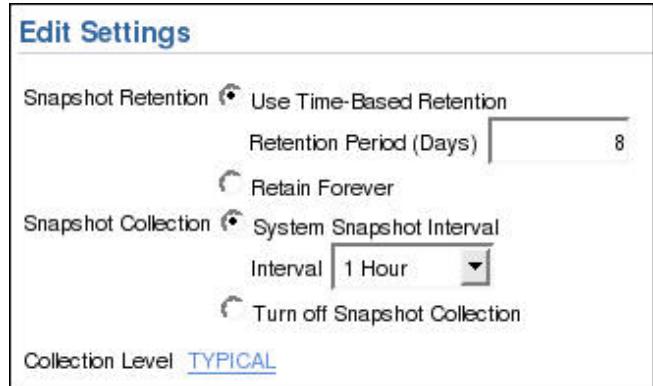
Server 탭을 누르고 Statistics Management 섹션에서 Automatic Workload Repository를 누릅니다.  
Automatic Workload Repository 페이지에서 Edit를 눌러 설정을 변경합니다.

Automatic Workload Repository 페이지에서 다음 작업을 수행할 수 있습니다.

- Workload Repository 설정 편집
- 생성된 스냅샷의 세부 정보를 살펴보고 수동으로 새 스냅샷 생성
- AWR Baseline 생성
- AWR 보고서 생성

# AWR 관리

- **Retention 기간**
  - 기본값: 8일
  - 필요한 저장 영역 고려
- **수집 간격**
  - 기본값: 60분
  - 필요한 저장 영역 및 성능 영향 고려
- **수집 레벨**
  - Basic(대부분의 ADDM 기능 비활성화)
  - Typical(권장)
  - All(스냅샷에 SQL 튜닝 정보 추가)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## AWR 관리

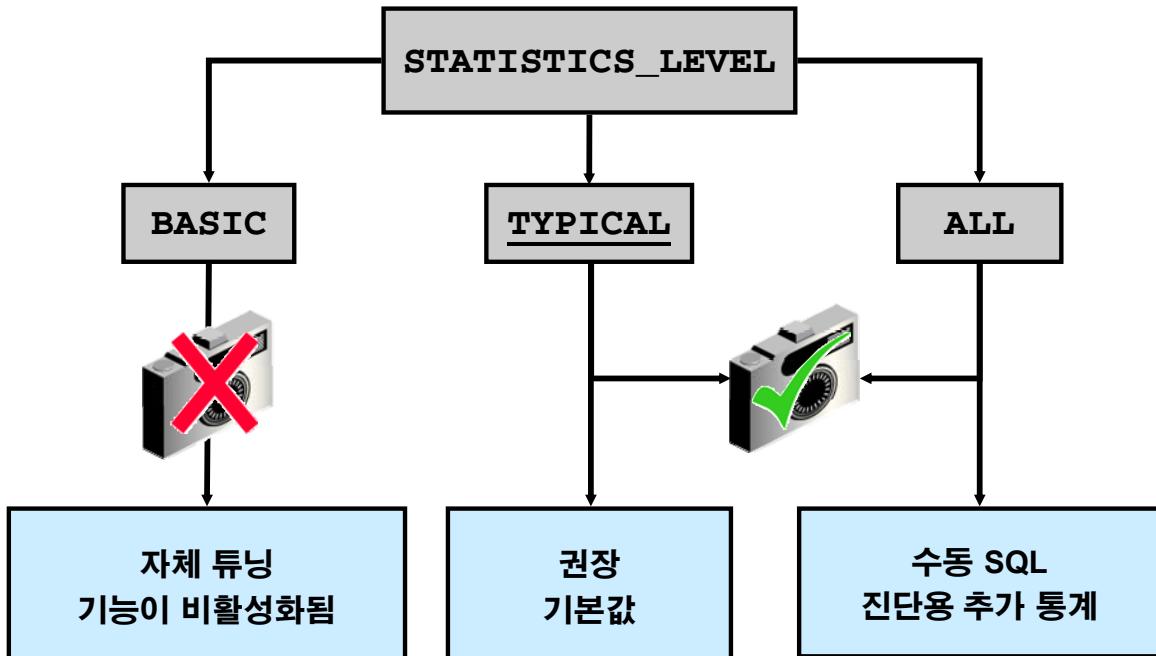
AWR 설정에는 Retention 기간, 수집 간격 및 수집 레벨(statistics\_level)이 포함됩니다. 이러한 설정을 임의로 줄이면 Advisor를 포함하여 AWR에 종속된 구성 요소의 기능이 떨어집니다.

설정을 늘리면 Advisor의 권장 기능이 향상되지만 대신 스냅샷 저장에 필요한 공간이 늘어나고 스냅샷 정보 수집에 소모되는 성능 요구 사항이 증가합니다.

새 응용 프로그램을 튜닝하는 경우에는 수집 레벨(statistics\_level)을 ALL로 설정해 보십시오.

ALL로 설정하면 SQL Advisor의 권장 기능을 향상시키는 SQL 실행 계획 및 타이밍 통계가 수집됩니다. 튜닝이 완료되면 이 설정을 TYPICAL 설정으로 되돌려야 합니다.

## 통계 레벨



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 통계 레벨

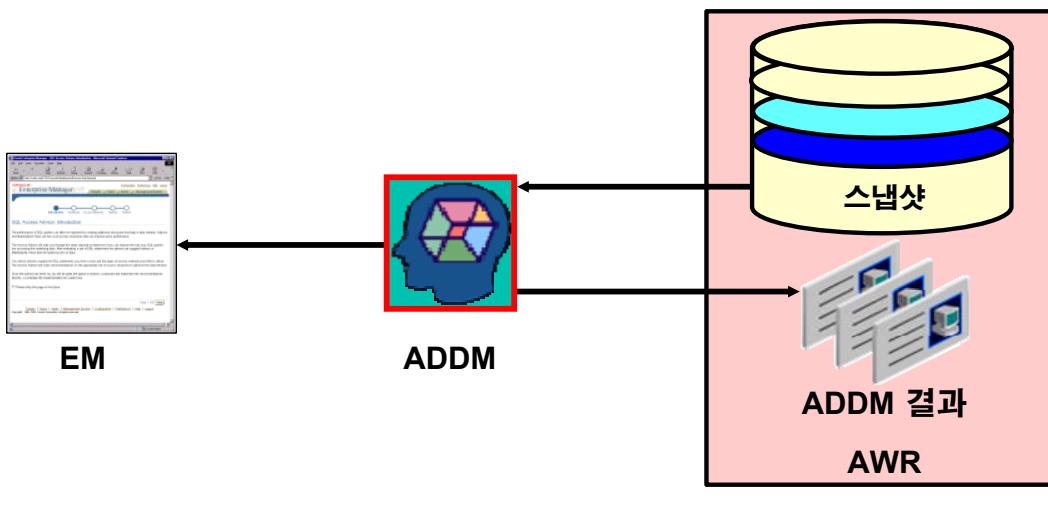
STATISTICS\_LEVEL 초기화 파라미터는 자동 유지 관리 업무를 포함하여 다양한 통계 및 Advisor 캡처를 제어합니다. 자동 유지 관리 업무에는 옵티마이저 통계 수집 업무가 포함됩니다. STATISTICS\_LEVEL 파라미터는 다음 레벨로 설정할 수 있습니다.

- **BASIC:** AWR 통계 및 metric이 계산되지 않습니다. 자동 옵티마이저 통계 업무와 모든 Advisor 및 Server-generated alert는 비활성화됩니다.
- **TYPICAL:** 데이터베이스 자체 관리에 필요한 주요 통계가 수집됩니다. 이러한 통계는 일반적으로 오라클 데이터베이스 동작을 모니터하는 데 필요한 항목을 나타냅니다. 여기에는 오래되거나 잘못된 통계로 인해 SQL 문의 성능이 저하될 가능성을 줄여주는 자동 통계 수집이 포함됩니다.
- **ALL:** 가능한 모든 통계가 캡처됩니다. 이 캡처 레벨을 사용하는 경우 시간이 지정된 OS 통계 및 계획 실행 통계가 추가됩니다. 이러한 통계는 대부분의 경우 필요하지 않으며 최적의 성능을 유지하려면 비활성화해야 합니다. 특정 진단 테스트의 경우 이러한 통계가 필요할 수도 있습니다.

오라클에서는 STATISTICS\_LEVEL 초기화 파라미터에 대해 기본값인 TYPICAL을 설정할 것을 권장합니다. 이 파라미터의 값을 BASIC으로 설정하면 옵티마이저 통계의 자동 수집이 비활성화됩니다.

# ADDM(Automatic Database Diagnostic Monitor)

- 각 AWR 스냅샷 생성 후마다 실행됩니다.
- instance를 모니터하여 병목 지점을 감지합니다.
- AWR에 결과를 저장합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ADDM(Automatic Database Diagnostic Monitor)

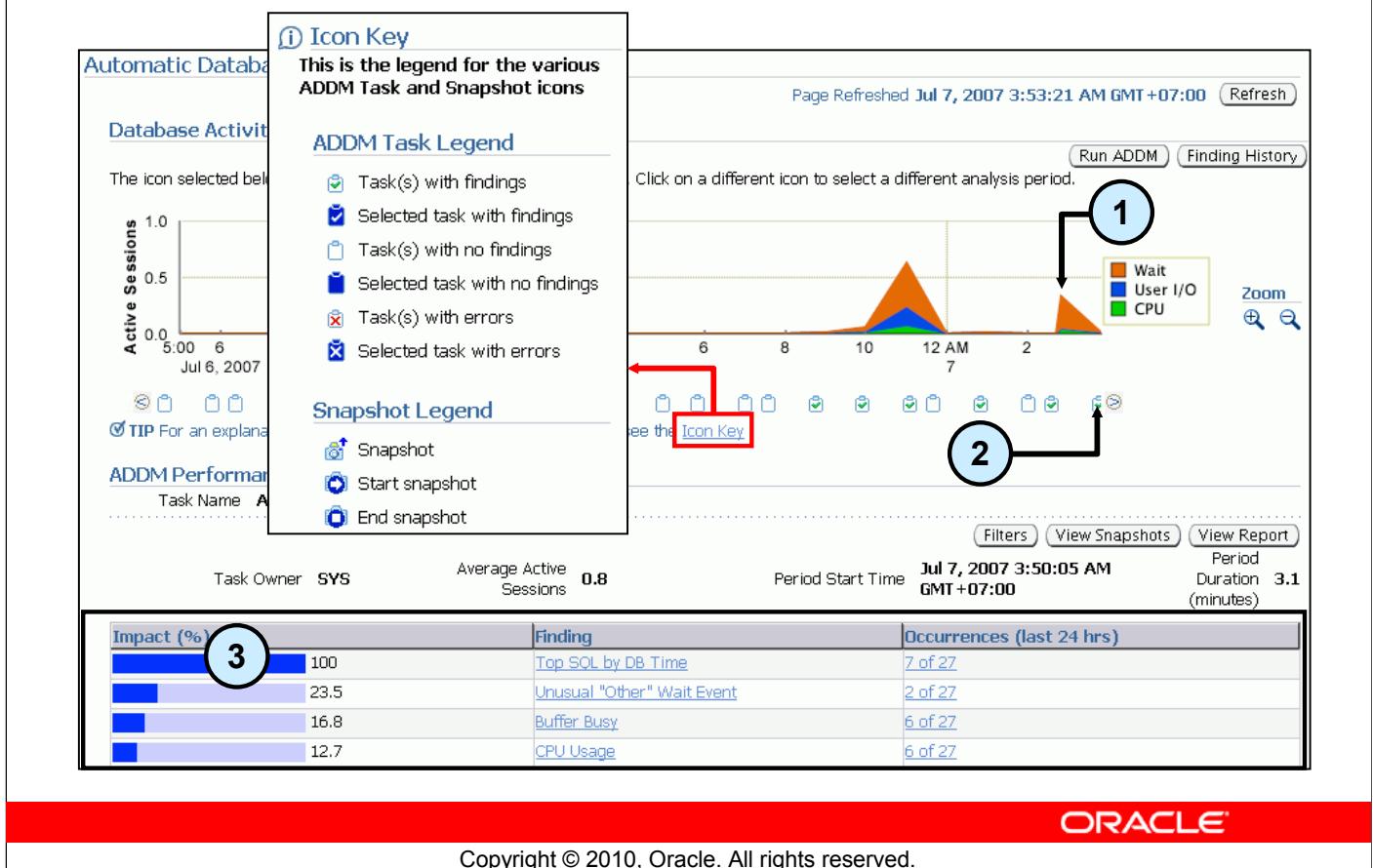
다른 advisor와 달리 자동 데이터베이스 진단 모니터(ADDM: Automatic Database Diagnostic Monitor)는 각 AWR 스냅샷 생성 후마다 자동으로 실행됩니다. 스냅샷이 생성될 때마다 ADDM은 마지막 두 스냅샷에 해당하는 기간을 분석합니다. ADDM은 Instance를 사전(Proactive)에 모니터하여 심각한 문제로 발전하기 전에 대부분의 병목 지점을 감지합니다. 대부분의 경우 ADDM은 감지된 문제의 해결 방법을 제시하며 이 방법으로 얻게 되는 이점을 수량화하여 보여주기도 합니다.

ADDM에 의해 감지되는 일반적인 문제는 다음과 같습니다.

- CPU 병목 지점
- 부실한 Oracle Net 연결 관리
- Lock 경합
- 입출력(I/O) 용량
- 너무 작은 데이터베이스 Instance 메모리 구조 크기
- 로드량이 많은 SQL 문
- 높은 PL/SQL 및 Java 시간
- 높은 체크포인트 로드 및 원인(예: 작은 로그 파일)

각 ADDM 분석의 결과는 AWR에 저장되며 Enterprise Manager를 통해서도 액세스할 수 있습니다.

# ADDM 결과



## ADDM 결과

Automatic Database Diagnostic Monitor (ADDM) 페이지에서 최근의 ADDM 실행에 대한 결과를 자세히 살펴볼 수 있습니다. Database Time은 분석 기간 동안 데이터베이스의 세션이 소비한 비휴지(non-idle) 시간의 합계를 나타냅니다. 발견된 각 사항마다 특정 영향 비율이 지정됩니다. 영향은 해당 문제에 소요된 시간을 분석 기간 동안의 데이터베이스 시간과 비교하여 나타냅니다. 슬라이드에서 다음을 살펴보십시오.

1. 그래프는 평균 활성 유저 수가 현재 시점에서 급격하게 증가했음을 보여줍니다. 또한 주요 문제는 Wait 문제입니다.
2. 아이콘은 페이지 아래에 표시된 ADDM 결과가 현재 시점의 결과임을 보여줍니다. 다른 아이콘을 누르면 과거 시점으로 이동하여 이전의 분석을 볼 수 있습니다.
3. 결과는 ADDM에서 튜닝 가능한 영역으로 발견한 사항을 요약하여 제공합니다. 특정 문제를 누르면 Performance Finding Details 페이지로 이동합니다.

View Report 버튼을 누르면 자세한 성능 분석 내용을 텍스트 보고서 형태로 볼 수 있습니다.

# ADDM 권장 사항

**Performance Finding Details: Buffer Busy**

Finding **Read and write contention on database blocks was consuming significant database time.** [Finding History](#)

Impact (Active Sessions) .14  
Impact (%)  16.8

Period Start Time Jul 7, 2007 3:50:05 AM GMT+07:00  
Period Duration (minutes) 3.1  
Filtered No [Filters](#)

**Recommendations**

Show All Details | Hide All Details

Details	Category	Benefit (%)
Action	Schema	16.8
Consider using ORACLE's recommended solution of automatic segment space management in a locally managed tablespace for the tablespace "TBSSPC" containing the TABLE "SPC.SPCT" with object ID 82664. Alternatively, you can move this object to a different tablespace that is locally managed with automatic segment space management.		
Database Object <a href="#">SPC.SPCT</a>		
► Show	Schema	16.8
► Show	Schema	16.8

**Findings Path**

Expand All | Collapse All

Findings	Impact (%)	Additional Information
▼ Read and write contention on database blocks was consuming significant database time.	16.8	
Wait class "Concurrency" was consuming significant database time.	17.4	



Copyright © 2010, Oracle. All rights reserved.

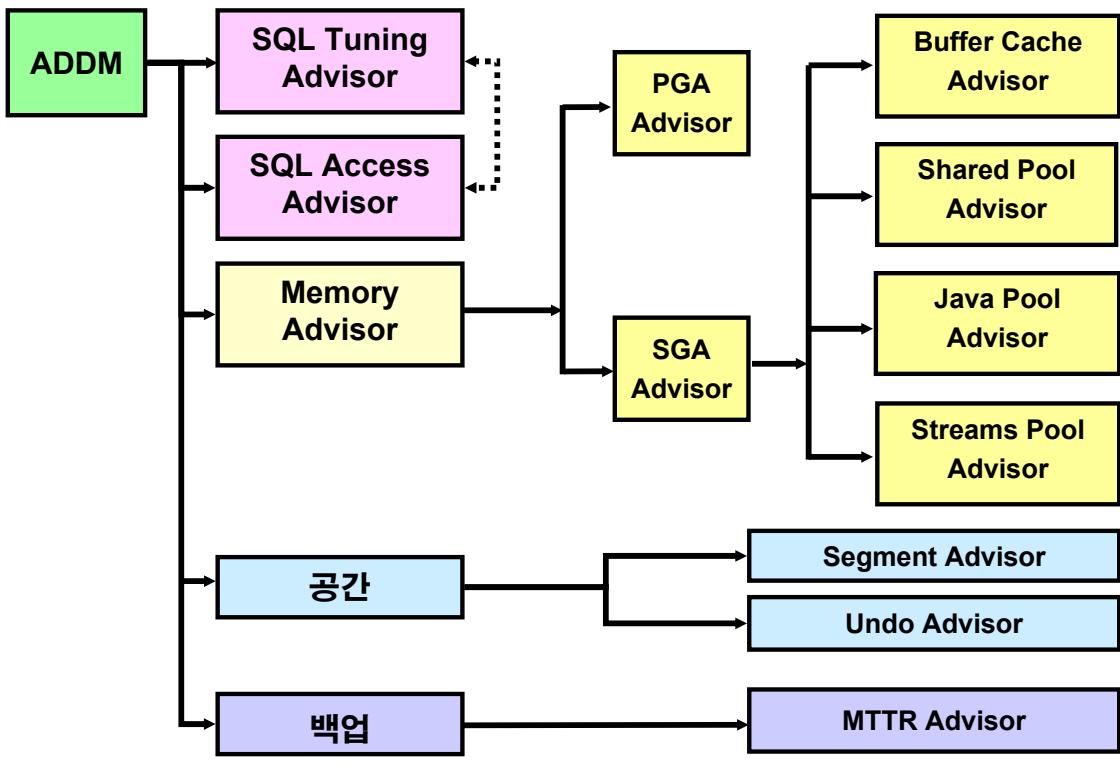
## ADDM 권장 사항

Performance Finding Details 페이지에는 해당 문제를 해결 할 수 있는 권장 사항이 표시됩니다. 권장 사항은 Schema, SQL Tuning, Database Configuration 등의 범주로 그룹화되어 있습니다. Benefit (%) 열은 권장 사항을 구현했을 때 최대한 줄일 수 있는 데이터베이스 경과 시간을 보여줍니다.

ADDM은 시스템에 대한 다양한 변경 사항을 고려합니다. 다음과 같은 권장 사항이 포함될 수 있습니다.

- **하드웨어 변경:** CPU 추가 또는 I/O 서브 시스템 구성 변경
- **데이터베이스 구성:** 초기화 파라미터 설정 변경
- **스키마 변경:** 테이블이나 인덱스에 Hash-partitioning 수행 또는 자동 세그먼트 공간 관리(ASSM: Automatic Segment Space Management) 사용
- **응용 프로그램 변경:** 시퀀스에 캐시 옵션 사용 또는 바인드 변수 사용
- **다른 Advisor 사용:** 로드량이 많은 SQL에 SQL Tuning Advisor 실행 또는 자주 사용되는 객체에 Segment Advisor 실행

# Advisory 프레임워크



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Advisory 프레임워크

Advisor는 개별 서버 구성 요소의 리소스 활용 및 성능에 대한 유용한 피드백을 제공합니다. 예를 들어, Memory Advisor는 오라클 데이터베이스 Instance에서 사용하는 총 메모리 양을 제어하는 MEMORY\_TARGET 초기화 파라미터에 대한 권장 값을 제공합니다.

ADDM은 AWR에 캡처된 데이터를 바탕으로 오라클 데이터베이스에서 자체 성능을 진단하고 식별된 문제를 해결하는 방법을 결정하도록 할 수 있습니다. ADDM은 각 AWR 통계를 캡처한 후에 자동으로 실행되며 잠재적으로 다른 Advisor를 호출할 수 있습니다.

다음은 Advisor Infrastructure가 제공하는 주요 장점입니다.

- 모든 Advisor가 동일한 인터페이스를 사용합니다.
- 모든 Advisor가 Workload Repository를 사용하여 공통 데이터 소스와 결과 저장 영역을 가집니다.

슬라이드에는 일부 Advisor만 나와 있습니다. 예를 들어, Data Recovery Advisor 및 SQL Repair Advisor는 표시되어 있지 않습니다.

### ADDM(Automatic Database Diagnostic Monitor)

ADDM은 60분마다 데이터베이스 성능을 검토하는 서버 기반 Expert입니다. ADDM의 목표는 발생 가능성이 있는 시스템 병목 지점을 조기에 감지하여 시스템 성능이 현저하게 떨어지기 전에 해결 방법을 제시하는 것입니다.

## Advisory 프레임워크(계속)

### Memory Advisor

실제로 Memory Advisor는 데이터베이스 Instance가 사용하는 총 메모리에 대한 가장 효율적인 설정을 결정할 수 있도록 하는 다양한 Advisory 기능의 모음입니다. SGA(System Global Area)에는 Shared Pool, 데이터베이스 버퍼 캐시, Java Pool 및 Streams Pool에 대한 Advisor 집합이 있습니다. Java Pool 및 Streams Pool Advisor는 EM Memory Advisor 페이지에 표시되지 않습니다. PGA(Program Global Area)용 Advisor도 있습니다. 이 advisor는 advisory 기능 이외에도 Large Pool 및 Java Pool에 대한 중앙 제어점을 제공합니다.

### MTTR(Mean-Time-To-Recover) Advisor

MTTR Advisor를 사용하여 데이터베이스에서 Instance Crash 후에 Recovery에 필요한 시간을 설정할 수 있습니다.

### Segment Advisor

이 Advisor는 필요한 공간보다 많은 공간을 소비하는 테이블 및 인덱스를 찾습니다. 그런 다음 테이블스페이스 또는 스키마 레벨에서 비효율적인 공간 소비가 있는지 검사하고 공간 소비를 줄일 수 있는 스크립트를 생성합니다.

### SQL Access Advisor

이 Advisor는 주어진 기간 내에 실행되는 모든 SQL 문을 분석하여 성능 향상에 도움이 되는 추가 인덱스 또는 Materialized view를 생성하도록 제안합니다.

### SQL Tuning Advisor

이 Advisor는 개별 SQL 문을 분석하여 성능 향상을 위한 권장 사항을 제공합니다. 명령문 재작성, Instance 구성 변경 또는 인덱스 추가와 같은 작업이 권장 사항에 포함될 수 있습니다. SQL Tuning Advisor는 직접 호출할 수 없으며 대신 Top SQL 또는 Top Sessions와 같은 도구 내에서 호출하여 성능에 높은 영향을 미치는 SQL 문을 최적화할 수 있습니다.

### Undo Management Advisor

Undo Management Advisor를 사용하여 주어진 Retention 기간을 지원하는 데 필요한 언두 테이블스페이스 크기를 결정할 수 있습니다. 언두 관리 및 Advisor 사용은 "언두 데이터 관리" 단원에서 다릅니다.

### Data Recovery Advisor

이 Advisor는 지속 Data Failure를 자동으로 진단하고, 유저에게 복구 옵션을 제공하며, 유저의 요청 시 복구를 실행합니다. Data Recovery Advisor는 MTTR(Mean time to recover)을 줄이고 자동화된 데이터 Recovery를 위한 중앙 집중식 도구를 제공하는 데 사용됩니다.

### SQL Repair Advisor

Automatic Diagnostic Repository에서 문제를 생성하는 심각한 오류로 인해 SQL 문이 실패한 후에는 SQL Repair Advisor를 실행합니다. Advisor는 명령문을 분석하며 대부분의 경우에는 패치를 사용해 명령문을 복구할 것을 권장합니다. 권장 사항을 구현하면 적용되는 SQL 패치는 Query 옵티マイ저가 이후의 실행에 대해 다른 실행 계획을 선택하도록 하여 Failure를 우회합니다. 이 작업은 SQL 문 자체를 변경하지 않고 수행됩니다.

# Enterprise Manager 및 Advisor

The screenshot shows the Oracle Enterprise Manager Advisor Central interface. At the top, there are two tabs: 'Advisors' (highlighted with a red box) and 'Checkers'. Below the tabs, a message says 'Page Refreshed Jun 7, 2007 2:26:15 PM CDT' with a 'Refresh' button. The main area is divided into sections:

- Advisors**: Contains links to ADDM, Memory Advisors, and SQL Advisors, along with links to Automatic Undo Management, MTTR Advisor, and SQL Performance Analyzer.
- Advisor Tasks**: A search bar with a 'Change Default Parameters' button. It includes fields for 'Advisory Type' (set to 'All Types'), 'Task Name', 'Advisor Runs' (set to 'Last Run'), and 'Status' (set to 'All'). A note below says: 'By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.'
- Results**: A table with columns 'Advisory Type' and 'Name'. One row is selected, showing 'ADDM' and 'ADDM:115240'. There are buttons for 'View Result', 'Delete', 'Actions', 'Re-schedule', and 'Go'.
- Checkers**: A section with a table of integrity check types: DB Structure Integrity Check, Data Block Integrity Check, Redo Integrity Check, Transaction Integrity Check, Undo Segment Integrity Check, and Dictionary Integrity Check. A 'Checkers' tab is highlighted with a red box.

At the bottom right of the interface is the ORACLE logo, and at the very bottom center is the copyright notice: 'Copyright © 2010, Oracle. All rights reserved.'

## Enterprise Manager 및 Advisor

Advisor Central 페이지는 모든 advisor의 기본 페이지입니다. Database Control Home 페이지의 Related Links list에서 Advisor Central 링크를 누르면 이 페이지에 액세스할 수 있습니다. 그러나 Database Control에서 이 경로를 통해서만 Advisor를 호출할 수 있는 것은 아니며 특정 컨텍스트에서 advisor에 대한 액세스가 이루어질 수도 있습니다.

Advisor Central 페이지의 Advisors 탭에서는 Workload Repository에 등록된 모든 Advisor 작업을 나열할 수 있습니다. 또한 미리 정의된 기간이나 advisor 유형별로 이 리스트를 필터링할 수도 있습니다.

Advisor Central 페이지의 Checkers 탭을 사용하면 다양한 데이터베이스 무결성 검사기의 일정을 잡을 수 있습니다. 모든 검사기 실행을 이름, 유형 또는 기간별로 나열할 수 있습니다.

몇몇 advisor는 "언두 데이터 관리", "성능 관리", "백업 및 Recovery 개념" 단원에서 더욱 자세히 설명합니다.

**주:** 미래의 모든 업무에 대한 기본 만료 기간(일)을 변경하려면 Change Default Parameters 페이지를 사용하십시오. 또한 이 페이지를 사용하여 일부 주요 Advisor의 파라미터를 변경할 수 있습니다.

## DBMS ADVISED 패키지

프로시저	설명
CREATE_TASK	Repository에 새 작업을 생성합니다.
DELETE_TASK	Repository에서 작업을 삭제합니다.
EXECUTE_TASK	작업 실행을 시작합니다.
INTERRUPT_TASK	현재 실행 중인 작업을 일시 중지합니다.
GET_TASK_REPORT	지정된 작업에 대해 텍스트 보고서를 생성 및 반환합니다.
RESUME_TASK	일시 중지된 작업을 재개합니다.
UPDATE_TASK_ATTRIBUTES	작업 속성을 갱신합니다.
SET_TASK_PARAMETER	작업 파라미터를 수정합니다.
MARK_RECOMMENDATION	하나 이상의 권장 사항을 승인, 거부 또는 무시로 표시합니다.
GET_TASK_SCRIPT	승인된 모든 권장 사항을 스크립트로 생성합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### DBMS ADVISED 패키지

DBMS ADVISED 패키지에는 모든 Advisor 모듈에 대한 모든 상수와 프로시저 선언이 포함되어 있습니다. 이 패키지를 사용하여 명령행을 통해 작업을 실행할 수 있습니다.

Advisor 프로시저를 실행하려면 ADVISOR 권한이 있어야 합니다. ADVISOR 권한이 있으면 Advisor 프로시저와 뷰 모두에 완전히 액세스할 수 있게 됩니다.

주: DBMS ADVISED 패키지에 있는 모든 프로시저에 대한 자세한 내용은 *Oracle Database PL/SQL Packages and Types Reference*를 참조하십시오.

## 퀴즈

옵티マイ저 통계 num\_rows에는 항상 테이블에 대한 실제 행 수가 반영됩니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

# 자동화된 유지 관리 업무

Autotask 유지 관리 프로세스에서는 다음 작업이 수행됩니다.

1. 유지 관리 윈도우(Maintenance window)가 열립니다.
2. Autotask 백그라운드 프로세스가 작업의 일정을 잡습니다.
3. Scheduler가 작업을 시작합니다.
4. Resource Manager가 Autotask 작업의 영향을 제한합니다.

기본 Autotask 유지 관리 작업:

- 옵티마이저 통계 수집
- Automatic Segment Advisor
- Automatic SQL Advisor



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 자동화된 유지 관리 업무

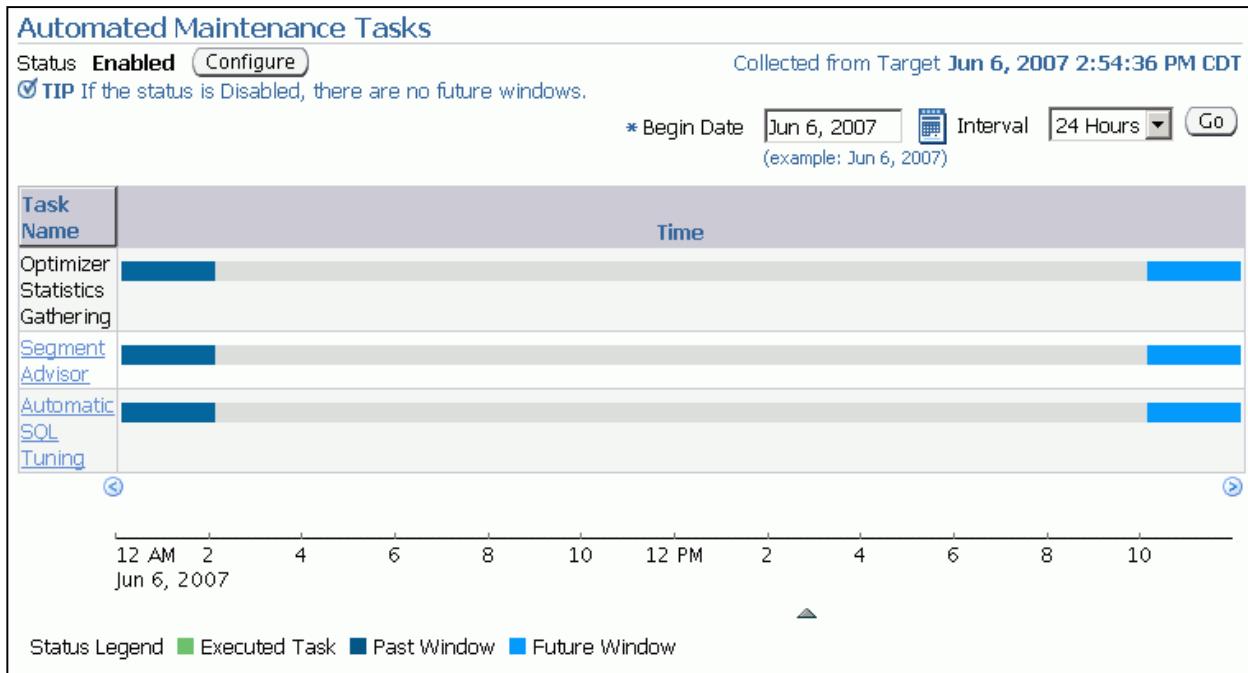
AWR에 저장된 정보를 분석함으로써 데이터베이스는 옵티마이저 통계 refresh와 같은 일상적인 유지 관리 업무가 필요한지 파악할 수 있습니다. 자동화된 유지 관리 업무 Infrastructure는 오라클 데이터베이스가 그러한 작업을 자동으로 수행할 수 있도록 활성화하며, Scheduler를 사용하여 미리 정의된 유지 관리 윈도우(Maintenance windows)에서 그러한 작업을 실행합니다.

기본적으로 주중 유지 관리 윈도우(Maintenance windows)는 오후 10:00에 시작되어 4시간 동안 지속됩니다. 토요일과 일요일에는 유지 관리 윈도우(Maintenance windows)가 오전 6:00에 시작되어 20시간 동안 유지됩니다. 시작 및 종료 시간, 빈도, 주중 요일 등을 비롯하여 유지 관리 윈도우(Maintenance windows)의 모든 속성은 커스터마이즈가 가능합니다. 또한 Database Resource Manager Resource Plan을 유지 관리 윈도우(Maintenance windows)에 연관시킴으로써 정상적인 데이터베이스 작업에 대해 자동화된 유지 관리 업무가 미치는 영향을 제한할 수 있습니다.

유지 관리의 예:

- 옵티마이저 통계는 자동 유지 관리 업무 Infrastructure를 사용하여 자동으로 새로 고쳐집니다.
- Automatic Segment Advisor에는 유지 관리 윈도우(Maintenance windows)에서 실행되는 기본 작업이 있습니다.
- DBCA로 데이터베이스를 생성할 때는 정기적인 데이터베이스 백업을 시작할 수 있습니다.

## 자동화된 유지 관리 업무



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 자동화된 유지 관리 업무(계속)

Server 페이지의 Scheduler 머리글 아래에서 Automated Maintenance Tasks를 누르면 Automated Maintenance Task 페이지에 액세스할 수 있습니다. 이 페이지에서 자동화된 유지 관리 업무 일정 및 최근 기록을 볼 수 있습니다. 여기서 일부 작업에 대한 세부 정보를 드릴 다운할 수 있습니다. Configure를 누르면 Automated Maintenance Tasks Configuration 페이지로 이동합니다. 작업은 윈도우에서 실행됩니다. 이 그래프에는 작업이 마지막으로 실행된 윈도우 및 작업이 실행되기로 일정이 잡힌 다음 윈도우가 표시됩니다.

**주:** 예제에는 작업의 기본 윈도우가 표시되어 있습니다. 유지 관리 윈도우(Maintenance windows)를 닫으면 Scheduler가 기본적으로 옵티마이저 통계 수집 작업을 종료합니다. 나머지 객체는 다음 번 유지 관리 윈도우(Maintenance windows)에서 처리됩니다.

## 자동화된 유지 관리 업무 구성

**Automated Maintenance Tasks Configuration**

Global Status  Enabled  Disabled

**Task Settings**

Optimizer Statistics Gathering  Enabled  Disabled [Configure](#)

Segment Advisor  Enabled  Disabled

Automatic SQL Tuning  Enabled  Disabled [Configure](#)

**Maintenance Window Group Assignment**

[Edit Window Group](#)

Window	Optimizer Statistics Gathering	Segment Advisor	Automatic SQL Tuning
<a href="#">WEDNESDAY WINDOW</a>	<a href="#">Select All</a> <a href="#">Select None</a>	<a href="#">Select All</a> <a href="#">Select None</a>	<a href="#">Select All</a> <a href="#">Select None</a>
<a href="#">THURSDAY WINDOW</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">FRIDAY WINDOW</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">SATURDAY WINDOW</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">SUNDAY WINDOW</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">MONDAY WINDOW</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">TUESDAY WINDOW</a>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Show SQL](#) [Revert](#) [Apply](#)

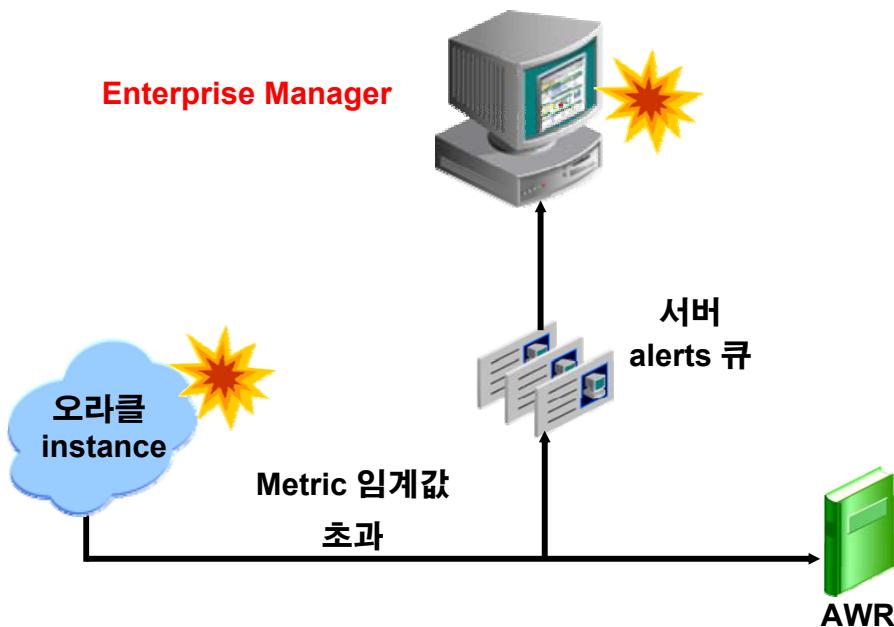
Copyright © 2010, Oracle. All rights reserved.

### 자동화된 유지 관리 업무 구성

Automated Maintenance Tasks Configuration 페이지에서는 자동 유지 관리 업무를 모두 한 번에, 개별 작업별로 또는 특정 윈도우별로 활성화 및 비활성화할 수 있습니다. 또한 옵티마이저 통계 수집에 사용되는 설정과 자동 SQL Tuning Advisor의 작업 제어 파라미터도 구성할 수 있습니다. 윈도우 이름을 선택하면 윈도우 스케줄을 보거나 편집할 수 있습니다.

Edit Window Group을 누르면 윈도우 그룹에서 윈도우를 추가하거나 제거할 수 있습니다.

## 서버 생성 Alert



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

### 서버 생성 Alert

Alert는 데이터베이스가 부적합한 상태에 있을 때 주의가 필요하다고 알려주는 통지입니다. 기본적으로 오라클 데이터베이스는 Enterprise Manager Database Control을 통해 Alert를 제공합니다. 선택적으로 Enterprise Manager에서 콘솔에 Alert 정보를 표시할 뿐만 아니라 문제 상황에 대한 전자 메일 메시지를 관리자에게 보내도록 구성할 수 있습니다.

또한 시스템에 대한 많은 관련 metrics에 임계값을 설정할 수 있습니다. Oracle Database 11g는 데이터베이스가 정상적인 읽기 상태에서 벗어나 이러한 임계값에 도달할 정도가 되면 사전 (Proactive)에 통지합니다. 발생 가능한 문제에 대한 조기 통지를 통해 신속하게 대처함으로써 대부분의 경우 유저가 이 문제를 발견하기 전에 해결할 수 있습니다.

기본적으로 다음을 포함하는 약 60개의 metrics가 모니터됩니다.

- Broken Job Count
- Database Time Spent Waiting (%)
- Dump Area Used (%)
- Baseline과 비교한 SQL Response Time (%)
- Tablespace Used (%)
- Generic Incident

다음과 같은 몇 가지 주요 metrics는 문제 통지를 조기에 제공할 수 있습니다.

- Average File Read Time (centiseconds)
- Response Time (per transaction)
- Wait Time (%)

## 임계값 설정

**Metric and Policy Settings**

**Metric Thresholds Policies**

**View Metrics with thresholds**

Metric	Comparison Operator	Warning Threshold	Critical Threshold	Corrective Actions	Collection Schedule	Edit
Access Violation	Matches		.*	None	Every 5 Minutes	
Access Violation Status	>		0	None	Every 5 Minutes	
Archive Area Used (%)	>	80		None	Every 15 Minutes	
Archiver Hung	Matches		.*	None	Every 5 Minutes	
Archiver Hung Status	>		0	None	Every 5 Minutes	
Audited User	=	SYS		None	Every 15 Minutes	
Average Users Waiting Count						
Administrative	>	10		None		
Application	>	10		None		
Cluster	>	30		None		
Commit	>	30		None		
Concurrency	>	10		None		
Configuration	>	10		None		
Network	>	10		None		

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 임계값 설정

전체 데이터베이스의 임계값을 설정하거나 편집하려면 Database Home 페이지의 Related Links 영역에서 "Metric and Policy Settings"를 누릅니다. 원하는 Warning 및 Critical 임계값을 입력합니다. 데이터베이스가 지정된 값에 도달하면 해당 Alert가 나타납니다.

이미 설정되어 있는 임계값은 "Metrics with thresholds" list에 표시됩니다. 기본적으로 약 60개의 metrics에 사전 설정된 임계값이 있으며, 필요한 경우 이를 변경할 수 있습니다. "All metrics" list에는 임계값이 설정되어 있지 않은 metrics가 표시됩니다.

Edit 아이콘 중 하나를 누르면 Warning 또는 Critical 임계값에 대한 추가 수정 조치를 지정할 수 있는 페이지에 액세스할 수 있습니다.

Collection Schedule 링크를 눌러 일정이 잡힌 수집 간격을 변경합니다. 각 일정은 metrics 그룹에 영향을 줍니다.

# Alert 생성 및 테스트

1. 임계값을 지정합니다.
2. 테스트 사례를 생성합니다.
3. Alert를 확인합니다.

The screenshot illustrates the three steps for creating and testing an alert:

- Step 1: Edit Advanced Settings**  
Shows the "Edit Advanced Settings: Tablespace Space Used (%)" dialog. A circled '1' highlights the "Warning Threshold" field set to 70. The dialog also includes tabs for "Monitored Objects" and "Comparison Operator".
- Step 2: Show SQL**  
Shows the SQL command used to create a table: `CREATE TABLE "HR"."FILLER" TABLESPACE "INVENTORY" STORAGE ( INITIAL 8M) AS SELECT * FROM HR.EMPLOYEES`.
- Step 3: Alerts**  
Shows the "Alerts" page with a table of triggered alerts. A circled '3' highlights the first row, which details a critical alert for a full tablespace named "INVENTORY".

ORACLE  
Copyright © 2010, Oracle. All rights reserved.

## Alert 생성 및 테스트

특정 객체에 대해 임계값을 설정할 수도 있습니다.

예

INVENTORY 테이블스페이스에서 사용되는 공간이 75%를 초과하면 Critical Alert를 받아야 하도록 결정합니다. 이 테이블스페이스에서는 해당 데이터 파일이 자동으로 확장되도록 허용하지 않습니다. Alert를 생성 및 테스트하려면 다음 단계를 수행합니다.

1. Enterprise Manager에서 "Metrics and Policy Settings" 페이지로 이동한 다음 Tablespace Used (%) 임계값에 대해 Edit 아이콘을 누릅니다. 테이블스페이스에 대해 원하는 임계값을 설정합니다.
2. Tables 페이지의 Schema 탭 아래에서 Alert를 테스트할 테이블을 생성합니다. "Define using SQL" 작업을 사용하여 기존 테이블을 복제(Duplicate)합니다. STORAGE 절의 초기 설정은 8MB이며, 이 설정을 사용하면 테이블이 10MB INVENTORY 테이블스페이스의 80%를 즉시 할당합니다.
3. 이 테이블을 확장할 수 없다는 오류가 나타나면 Database Home 페이지에서 연관된 Alert를 확인합니다. Tablespace Space Used (%)는 기본적으로 10분마다 수집됩니다.

대부분의 Alert에는 실행되면 자세한 권고를 보여주는 관련 Advisor의 이름이 포함되어 있습니다. 해당하는 각 alert 메시지에 대해 Database Control은 관련된 advisor를 실행하는 링크를 제공합니다.

# Alert 통지

Select Metric ▲	Objects	Severity States	Corrective Action States		Edit
			On Critical	On Warning	
<input type="checkbox"/> Session Limit Usage (%)	n/a	Critical			
<input type="checkbox"/> Session Terminated Status	n/a	Critical			
<input type="checkbox"/> Tablespace Space Used (%)	All Objects (Tablespace Name)	Critical			
<input type="checkbox"/> Wait Time (%)	n/a	Critical			

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## Alert 통지

통지 메커니즘은 Enterprise Manager 유저 인터페이스를 사용합니다. 통지 메커니즘의 개념은 앞으로 발생할 일련의 Alert에 적절한 통지 메커니즘을 수립하는 통지 규칙을 기반으로 합니다.

Database Control을 사용하여 통지 규칙을 편집할 수 있습니다. Home 페이지에서 Preferences 링크를 누르면 General 페이지가 표시됩니다. 이 페이지에서 통지를 받을 전자 메일 주소를 지정합니다.

General 페이지의 Notification 영역에서 Rules 링크를 누릅니다. "Database Availability and Critical States" 규칙을 선택한 다음 Edit 버튼을 누릅니다. 그러면 "Edit Notification Rule Database Availability and Critical States" 페이지가 나타나며, 이 페이지에서 Metrics 탭을 눌러 통지를 수신할 Metrics를 편집할 수 있습니다.

## Alert 통지(계속)

선택적으로 특정 Alert가 발생하면 Enterprise Manager가 직접 통지를 제공하도록 지정할 수 있습니다. 예를 들어, Critical Alert에 대해 전자 메일 통지를 보내도록 지정하고 각 호출 metric마다 시스템 응답 시간에 대해 Critical 임계값이 설정된 경우 다음과 유사한 메시지를 포함하는 전자 메일을 보낼 수 있습니다.

```
Host Name=mydb.us.mycompany.com
Metric=Response Time per Call
Timestamp=08-NOV-2005 10:10:01 (GMT -7:00)
Severity=Critical
Message=Response time per call has exceeded the threshold.
See the latest ADDM analysis.
Rule Name= Rule
Owner=SYSMAN
```

전자 메일에는 호스트 이름과 가장 최근의 ADDM 분석에 대한 링크가 포함됩니다.

기본적으로 DB Down, Generic Alert Log Error Status, Tablespace Used와 같은 Critical 상태의 Alert는 통지 대상으로 설정됩니다. 그러나 이러한 통지를 수신하려면 다음 단계를 수행하여 전자 메일 정보를 설정해야 합니다.

1. Database Control 페이지에서 머리글/바닥글 영역의 Setup 링크를 누릅니다.
2. Setup 페이지에서 Notification Methods를 선택합니다.
3. Notifications Methods 페이지의 Mail Server 영역에 필요한 정보를 입력합니다.

스크립트 및 SNMP(Simplified Network Management Protocol) 트랩을 비롯한 다른 통지 방법이 있습니다. SNMP 트랩은 Third-party 응용 프로그램과의 통신에 사용할 수 있습니다.

통지를 수신하려면 다음과 같이 하십시오.

1. Database Control 페이지에서 머리글/바닥글 영역의 Preferences 링크를 누릅니다.
2. Preferences 페이지에서 General을 선택합니다. E-mail Addresses 영역에 전자 메일 주소를 입력합니다.
3. 선택적으로 수신하는 통지에 대한 심각도 상태 변경과 같은 통지 규칙을 편집할 수 있습니다. 이렇게 하려면 Notification Rules를 누릅니다. Notification Rules 페이지가 나타납니다.  
주: 통지 규칙 구성에 대한 자세한 사항은 *Oracle Enterprise Manager Advanced Configuration* 설명서를 참조하십시오.

## Alert에 대한 대응

- 필요한 경우 ADDM 또는 다른 Advisor를 실행하는 등의 방법으로 더 많은 입력을 수집해야 합니다.
- Critical 오류를 조사합니다.
- 수정 조치를 취합니다.
- 자동으로 지워지지 않는 Alert의 경우 직접 확인합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Alert에 대한 대응

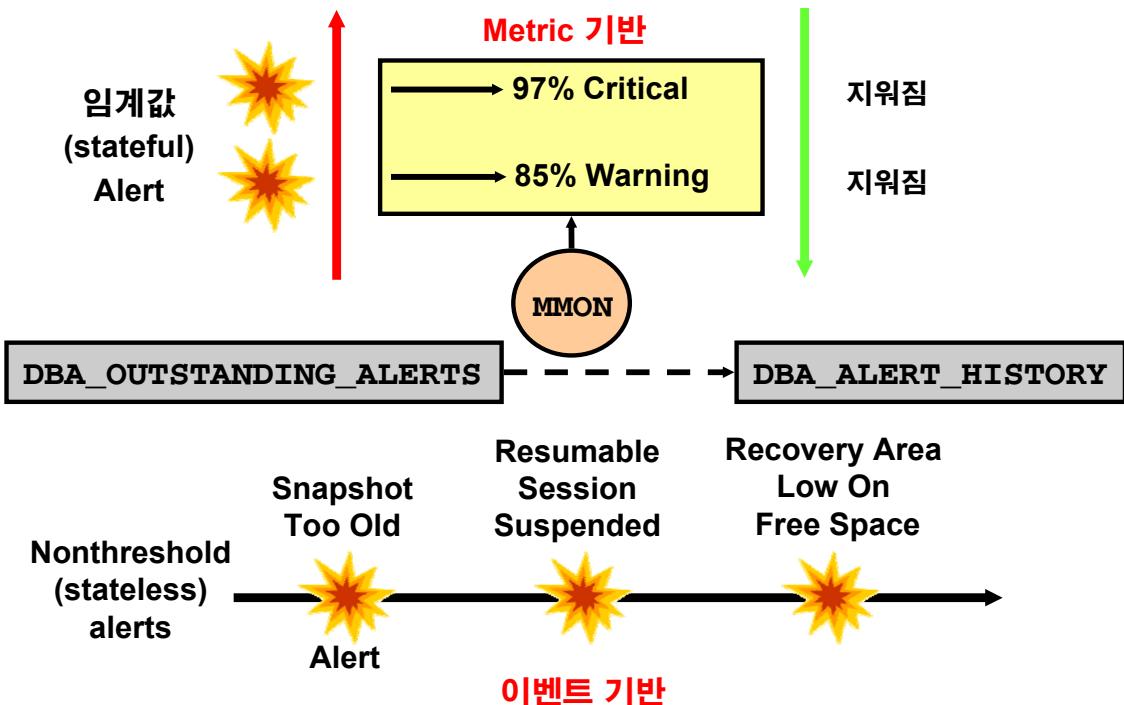
Alert를 수신하면 해당 Alert에서 제공하는 권장 사항을 따릅니다. 또는 ADDM이나 기타 적절한 Advisor를 실행하여 시스템 또는 객체 동작에 대한 보다 자세한 진단을 확인할 수도 있습니다.

Critical 오류에 대해서는 Alert 및 Incident가 생성됩니다. 일반적으로 Critical 오류는 문제로 수집되는 Incident를 생성합니다. Support Workbench를 사용하여 문제를 조사하고 가능한 경우 Oracle 지원 센터로 보고합니다.

"Out of Space"와 같은 대부분의 Alert는 문제의 원인이 사라지면 자동으로 지워집니다. 그러나 Generic Alert Log Error와 같은 Alert는 유저에게 통지되며 유저가 반드시 확인해야 합니다. 필요한 수정 조치를 취한 후에 Alert를 지우거나 삭제하여 Alert를 확인합니다. Alert를 지우면 Home 페이지의 Related Links 아래에 표시되는 Alert History로 Alert가 보내집니다. Alert를 삭제하면 Alert History에서 제거됩니다.

Generic Alert Log Error와 같은 Alert를 지우려면 Home 페이지의 Diagnostic Summary에서 Alert Log 링크를 누릅니다. Alert Log Errors 페이지가 나타납니다. 지울 Alert를 선택하고 Clear를 누릅니다. Alert를 삭제하려면 삭제할 Alert를 선택하고 Purge를 누릅니다. Clear Every Open Alert 버튼 또는 Purge Every Alert 버튼을 눌러도 됩니다.

## Alert 유형 및 Alert 지우기



ORACLE

Copyright © 2010, Oracle. All rights reserved.

### Alert 유형 및 Alert 지우기

Server-generated alert에는 임계값과 비임계값의 두 가지 종류가 있습니다.

대부분의 서버 생성 alert는 데이터베이스 metric에 대해 warning 및 critical 임계값을 설정하여 구성됩니다. 다음을 포함하여 120개가 넘는 metrics에 대한 임계값을 정의할 수 있습니다.

- Physical Reads Per Sec
- User Commits Per Sec
- SQL Service Response Time

데이터베이스와 관련된 Tablespace Space Usage metric을 제외한 다른 metric은 Instance와 관련이 있습니다. 임계값 Alert는 *stateful alerts*라고도 하며, Alert 조건이 지워지면 자동으로 지워집니다.

Stateful alert는 DBA\_OUTSTANDING\_ALERTS에 나타나며 지워지면 DBA\_ALERT\_HISTORY로 이동합니다.

다른 서버 생성 Alert는 ORA-\* 오류, "Snapshot Too Old" 오류, Recovery Area Low On Free Space 및 Resumable Session Suspended와 같은 특정 데이터베이스 이벤트에 해당합니다. 이러한 alert는 비임계값 기반 alert로 *stateless alert*라고도 합니다. Stateless Alert는 기록 테이블로 직접 이동됩니다. Database Control은 Stateless alert를 자신의 repository에 저장하므로 Stateless alert는 Database Control 환경에서만 지울 수 있습니다.

## 퀴즈

**SNAPSHOT TOO OLD와 같은 Stateless alert는 딕셔너리  
뷰 DBA\_OUTSTANDING\_ALERTS에서 찾을 수 있습니다.**

- 1. 참**
- 2. 거짓**

**ORACLE®**

Copyright © 2010, Oracle. All rights reserved.

**해답: 2**

## 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 옵티마이저 통계 관리
- AWR(Automatic Workload Repository) 관리
- ADDM(Automatic Database Diagnostic Monitor) 사용
- advisory 프레임워크 설명 및 사용
- alert 임계값 설정
- Server-Generated Alert 사용
- 자동화된 업무 사용

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 연습 12 개요: Proactive Maintenance

이 연습에서는 다음을 포함하여 ADDM을 사용한 데이터베이스 Proactive Maintenance에 대해 다룹니다.

- 분석을 위한 문제 설정
- 데이터베이스 성능 검토
- 해결 방안 구현



Copyright © 2010, Oracle. All rights reserved.



# 13

성능 관리

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## 목표

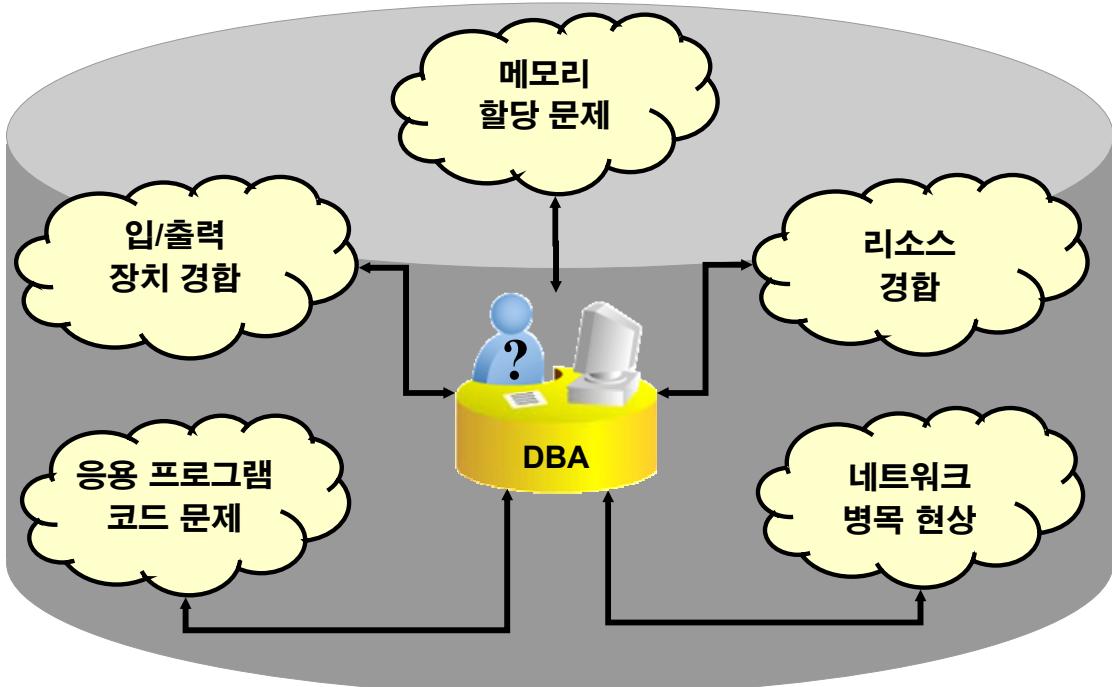
이 단원을 마치면 다음을 수행할 수 있습니다.

- Enterprise Manager를 사용하여 성능 모니터
- AMM(자동 메모리 관리) 사용
- Memory Advisor를 사용하여 메모리 버퍼의 크기 조정
- 성능 관련 동적 뷰 보기
- invalid 및 unusable 객체 문제 해결



Copyright © 2010, Oracle. All rights reserved.

## 성능 모니터



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

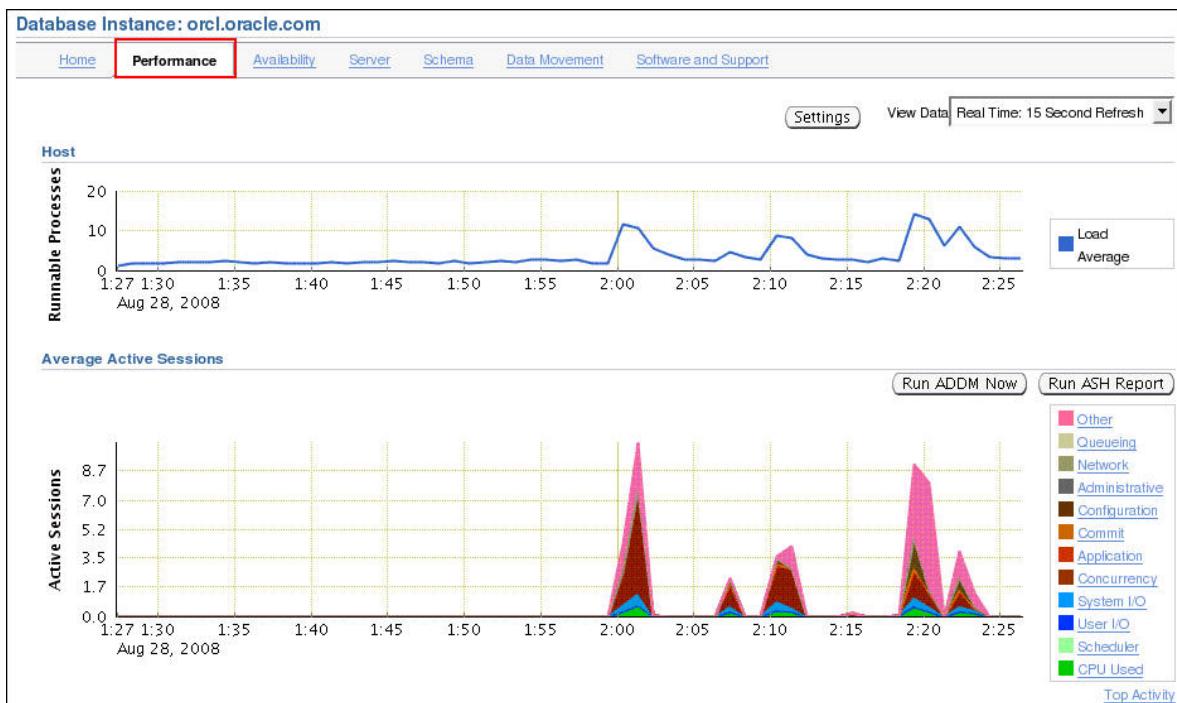
### 성능 모니터

Oracle Database 11g를 관리하고 원활하게 실행되도록 하려면 데이터베이스 관리자(DBA)는 정기적으로 성능을 모니터하여 병목 지점을 찾고 문제가 되는 영역을 수정해야 합니다.

DBA는 네트워크 성능에서 디스크 입/출력(I/O) 속도, 개별 응용 프로그램 작업에 걸린 시간까지 다양한 성능 측정을 수행할 수 있습니다. 이러한 성능 측정 단위를 일반적으로 *데이터베이스 metrics*라고 합니다.

주: 오라클 데이터베이스 성능에 대한 자세한 사항은 *Oracle Database 11g: Performance Tuning* 과정을 참조하십시오.

# Enterprise Manager Performance 페이지



ORACLE®

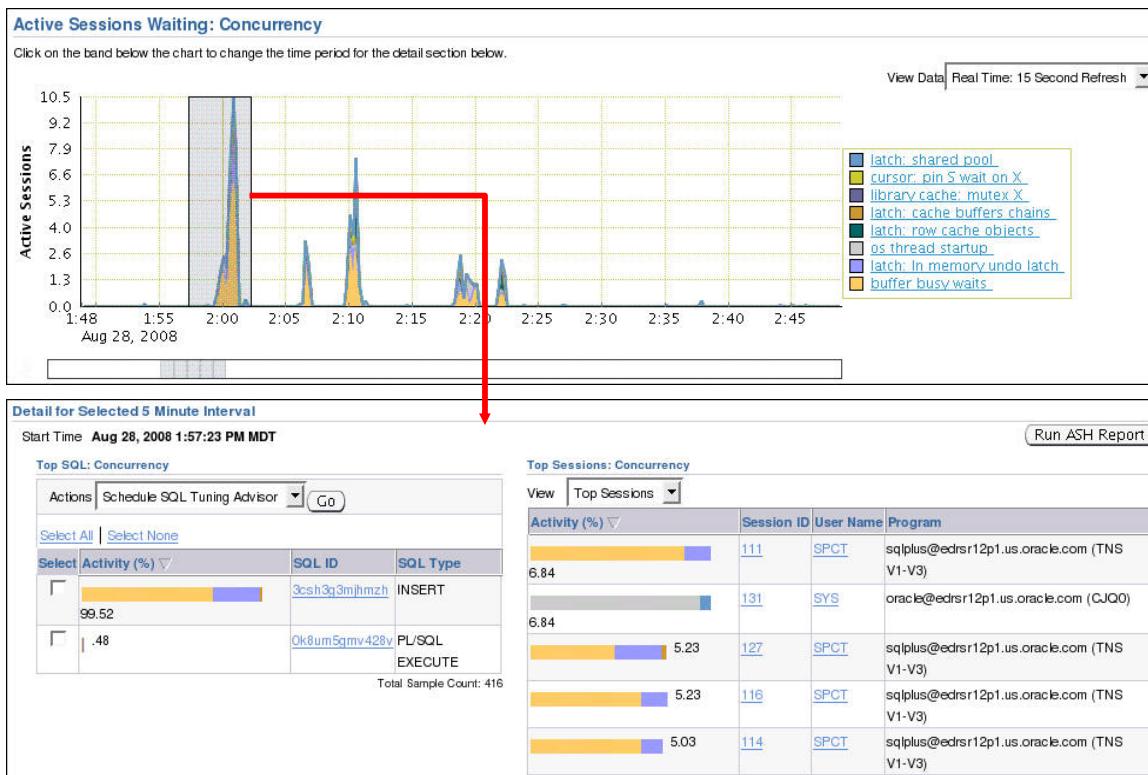
Copyright © 2010, Oracle. All rights reserved.

## Enterprise Manager Performance 페이지

Enterprise Manager의 Performance 페이지에서는 강력한 기능의 여러 가지 성능 모니터 및 튜닝 도구를 모두 사용할 수 있습니다. 이 페이지의 첫번째 그래프 집합에는 요약된 프로세스 및 활성 세션 작업이 표시됩니다. Average Active Sessions 그래프는 CPU 사용량 레벨을 표시하고 어떤 리소스가 대부분의 대기 이벤트를 발생시키는지 보여줍니다.

슬라이드에서는 Concurrency 및 Other 대기 시간이 최근 증가한 것을 볼 수 있습니다. 이러한 각 스파이크 중에는 시스템 I/O와 CPU 사용량도 약간 증가했습니다. 이러한 범주를 누르면 대기에 대한 세부 정보를 볼 수 있습니다. I/O 데이터는 로그 파일 읽기, 콘트롤 파일 쓰기 등과 같은 I/O 유형으로 세분화됩니다.

## 특정 대기 범주로 드릴 다운



**ORACLE**

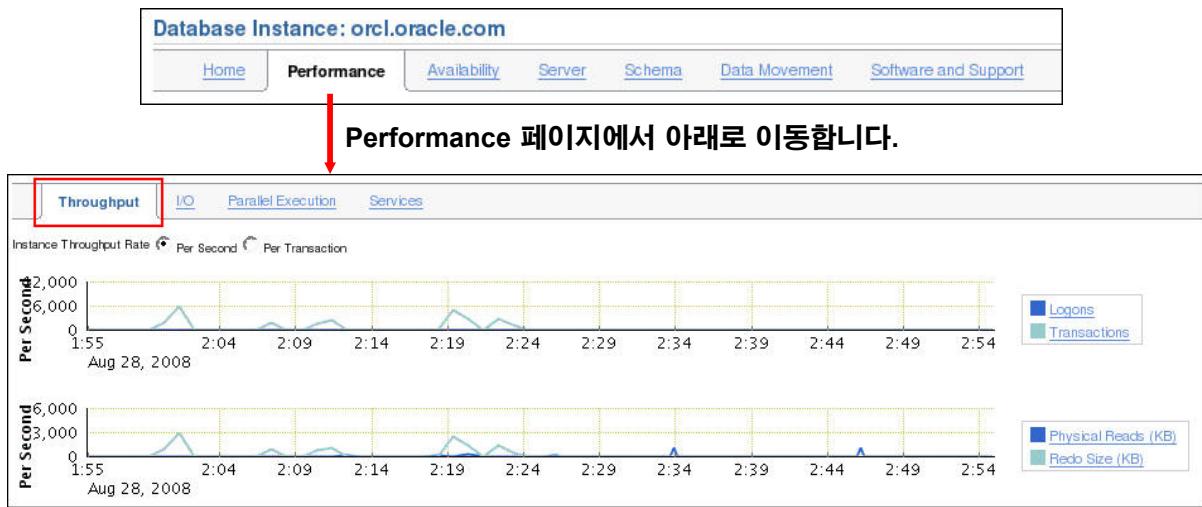
Copyright © 2010, Oracle. All rights reserved.

### 특정 대기 범주로 드릴 다운

특정 대기 범주를 드릴 다운하면 5분 간격으로 세부 정보를 볼 수 있고 또한 이 시간 동안 해당 대기 이벤트와 연관된 Top Working SQL 및 Top Working Sessions를 볼 수 있습니다. 이렇게 함으로써 시스템 성능 저하(slowdown)에 대한 사후 분석을 수행하고 잠재적인 원인을 파악할 수 있습니다.

슬라이드의 예제는 이전 슬라이드의 Active Session 그래프에서 Concurrency 범주로 드릴 다운한 결과를 보여 줍니다.

# Performance 페이지: 처리 능력



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

## Performance 페이지: 처리 능력

Performance 페이지의 Throughput 및 I/O 탭을 누르면 Instance 처리 능력 및 Instance 디스크 I/O그래프를 볼 수 있습니다. 이 슬라이드에서는 Throughput 탭이 선택되어 있습니다.

# 성능 모니터: Top Sessions

**Top Consumers**

Collected From Jul 31, 2009 8:40:17 AM GMT+07:00 To Jul 31, 2009 8:40:32 AM GMT+07:00

Overview Top Services Top Modules Top Actions Top Clients Top Sessions

Kill Session View Disable SQL Trace Enable SQL Trace

Select	SID	DB User	CPU (1/100 sec)	PGA Memory (bytes)	Physical Reads	Logical Reads	Hard Parses	Total Parses	Disk Sorts	Status	Program	Module	OS PID	Machine	OS User	SQL Trace
(36)	36	INVENTORY	12	1540712	20	228	0	0	0	ACTIVE	sqlplus@edrsr12p1.us.oracle.com	SQL*Plus	18269	edrsr12p1.us.oracle.com	oracle	DISABLED
(59)	59	DBSNMP	12	7168500	0	56	0	10	0	ACTIVE	emagent@edrsr12p1.us.oracle.com	emagent_SQL_database	9753	edrsr12p1.us.oracle.com	oracle	DISABLED
(42)	42	CJQ0	0	1794548	0	6	0	0	0	ACTIVE	oracle@edrsr12p1.us.oracle.com	(CJQ0)	13001	edrsr12p1.us.oracle.com	oracle	DISABLED
(41)	41	DBSNMP	0	2384372	0	3	0	2	0	ACTIVE	emagent@edrsr12p1.us.oracle.com	emagent_AQMetrics	13286	edrsr12p1.us.oracle.com	oracle	DISABLED
(29)	29	SYSMAN	0	2843124	0	1	0	5	0	ACTIVE	OMS	OEM.DefaultPool	12965	edrsr12p1.us.oracle.com	oracle	DISABLED
(49)	49	DBSNMP	14	1532404	0	0	0	2	0	ACTIVE	OMS	Realtime Connection	18390	edrsr12p1.us.oracle.com	oracle	DISABLED
(34)	34	SYSMAN	0	2384372	0	0	0	0	0	ACTIVE	OMS	OEM.SystemPool	13061	edrsr12p1.us.oracle.com	oracle	DISABLED
(30)	30	SYSMAN	0	3105268	0	0	0	3	0	ACTIVE	OMS	OEM.DefaultPool	12967	edrsr12p1.us.oracle.com	oracle	DISABLED
(18)	18	MMNL	0	1270260	0	0	0	0	0	ACTIVE	oracle@edrsr12p1.us.oracle.com	(MMNL)	12861	edrsr12p1.us.oracle.com	oracle	DISABLED
(25)	25	QMNC	0	745972	0	0	0	0	0	ACTIVE	oracle@edrsr12p1.us.oracle.com	STREAMS	12963	edrsr12p1.us.oracle.com	oracle	DISABLED
(57)	57	J001	0	418292	0	0	0	0	0	ACTIVE	oracle@edrsr12p1.us.oracle.com	(J001)	18451	edrsr12p1.us.oracle.com	oracle	DISABLED
(43)	43	SMCO	0	418292	0	0	0	0	0	ACTIVE	oracle@edrsr12p1.us.oracle.com	KTSJ	13544	edrsr12p1.us.oracle.com	oracle	DISABLED

Copyright © 2010, Oracle. All rights reserved.

## 성능 모니터: Top Sessions

Additional Monitoring Links 섹션에서 Top Consumers를 누르면 Top Consumers 페이지로 이동합니다.

Top Consumers Overview 페이지에는 다음과 같은 항목이 그래픽 형식으로 표시됩니다.

- Top Services
- Top Modules(서비스별)
- Top Actions(서비스 및 모듈별)
- Top Clients

Top Consumers 페이지에서 Top Sessions 탭을 누르면 가장 많은 리소스를 사용하는 세션에 대해 다음과 같은 critical 통계를 볼 수 있습니다.

- CPU
- PGA Memory
- Logical Reads
- Physical Reads
- Hard Parses
- Disk Sorts

해당 열에서 결과를 값별로 정렬할 열 이름을 누릅니다.

이 페이지의 테이블에는 Logical Reads를 기준으로 세션이 정렬되어 있습니다. 테이블을 살펴보면 특정 시간에 세션 36의 유저 INVENTORY가 가장 많은 Logical Reads를 생성한다는 것을 알 수 있습니다.

## 성능 모니터: Top Services

The screenshot shows the 'Top Services' tab selected in the navigation bar. The main content area displays a table of services and their activity over the last 5 minutes. A second table below provides detailed activity metrics for each service.

Select	Service	Activity (% for the last 5 minutes)	SQL Trace Enabled	Delta Elapsed Time (seconds)	Cumulative Elapsed Time (seconds)
<input type="checkbox"/>	SYS\$USERS	42.9	FALSE	0	227
<input type="checkbox"/>	SYS\$BACKGROUND	35.7	FALSE	0	0
<input type="checkbox"/>	SH	14.3	FALSE	0	2
<input type="checkbox"/>	SERV1	7.1	FALSE	0	2

	Delta CPU Time (seconds)	Cumulative CPU Time (seconds)	Delta Physical I/O (blocks)	Cumulative Physical I/O (blocks)
	0	0	0	16031
	0	137	0	14414
	0	1	15	637
	0	2	0	12

ORACLE

Copyright © 2010, Oracle. All rights reserved.

### 성능 모니터: Top Services

하나의 Application Server가 데이터베이스 연결을 풀링하고 있는 다계층 시스템에서는 세션을 보는 것만으로는 성능 분석에 필요한 정보를 얻을 수 없습니다. 세션을 서비스 이름으로 그룹화하면 성능을 보다 정확하게 모니터할 수 있습니다.

슬라이드 예제에는 SYS\$USERS, SYS\$BACKGROUND, SH 및 SERV1의 4개 서비스가 있습니다. 특정 요청에 어떤 세션이 사용되었는지에 관계없이, 이러한 서비스 중 하나를 통해 연결되었으면 세션의 성능 데이터가 해당 서비스 이름으로 캡처됩니다. 슬라이드에서 응용 프로그램 서비스(SH 및 SERV1)에 대해 나열되어 있는 내용을 볼 때 지난 5분 동안 SH 서비스가 보다 많은 작업을 수행했음을 확인하게 알 수 있습니다.

## 메모리 구성 요소 관리

- **AMM(자동 메모리 관리)**
  - Instance에 할당되는 총 메모리를 지정할 수 있습니다 (SGA 및 PGA 모두 포함).
- **자동 공유 메모리 관리(Automatic Shared Memory Management):**
  - 하나의 초기화 파라미터를 통해 총 SGA 메모리를 지정할 수 있습니다.
  - Oracle 서버가 Shared Pool, Java Pool, 버퍼 캐시, Streams Pool 및 Large Pool에 할당된 메모리 양을 관리할 수 있습니다.
- **수동으로 공유 메모리 관리 설정:**
  - 여러 개별 초기화 파라미터를 통해 구성 요소의 크기를 조정합니다.
  - 적합한 Memory Advisor를 사용하여 권장 사항을 제공합니다.

**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 메모리 구성 요소 관리

Oracle Database 11g에서는 Instance에 할당되는 총 메모리를 지정할 수 있습니다. 메모리는 필요에 따라 SGA(System Global Area) 및 PGA(Program Global Area) 간에 동적으로 재할당됩니다. 이 방식을 AMM(자동 메모리 관리)이라고 하며, 동적 메모리 해제를 지원하는 플랫폼에서만 사용할 수 있습니다. 따라서 메모리 관리 작업이 간편해집니다.

Memory Advisor를 사용하면 다양한 레벨에서 초기화 파라미터를 설정할 수 있습니다. 사용 가능한 Advisor는 메모리 파라미터를 지정하는 레벨에 따라 다릅니다. AMM을 활성화하는 경우에는 Memory Size Advisor를 사용할 수 있습니다.

ASMM(자동 공유 메모리 관리)을 사용하면 SGA를 전체적으로 관리할 수 있습니다. SGA는 여러 구성 요소로 이루어집니다. 이러한 대부분의 구성 요소 크기는 초기화 파라미터의 한도 내에서 최적의 성능을 낼 수 있도록 동적으로 조정됩니다. AMM을 활성화하면 ASMM도 자동으로 활성화됩니다. AMM은 활성화하지 않고 ASMM만 활성화하는 경우에는 SGA Size Advisor를 사용할 수 있습니다.

각 구성 요소에 대해 초기화 파라미터를 설정하면 개별 구성 요소의 크기를 수동으로 관리할 수 있습니다. Oracle 서버가 SGA 또는 PGA 구성 요소의 크기와 관련된 성능 문제를 통지하는 경우에는 해당 구성 요소의 Memory Advisor를 사용하여 적절한 새 설정을 결정할 수 있습니다. Memory Advisor는 파라미터 변경이 주는 영향을 모델링할 수 있습니다.

# AMM(자동 메모리 관리) 활성화

The screenshot shows the Oracle Database Advisor Central interface. On the left, under 'Memory Advisors', there is a note: 'Enable을 눌러 자동 메모리 관리를 활성화합니다.' (Press Enable to activate automatic memory management). Below this, the 'Automatic Memory Management' status is shown as 'Disabled' with a red box around the 'Enable' button. A yellow arrow points from this note to the 'Enable' button.

On the right, a graph titled 'Memory Size Advice' plots 'Improvement in DB Time (%)' against 'Total Memory Size (MB)'. The graph shows a blue curve with a green vertical line at approximately 556 MB. A legend indicates: 'Percentage improvement in DB Time for various sizes of Total Mem' (blue line), 'Total Memory Size' (green cross), and 'Maximum Memory Size' (red dot). The 'Total Memory Size (MB)' is set to 556. The graph has a note: 'You can click on the curve in the graph to set a new value. Total Memory Size cannot be greater than the Maximum Memory Size. First modify the Maximum Memory size (from the parent page) and then select a value of Total Memory up to the Maximum Memory size.'

At the bottom right of the interface is the ORACLE logo.

## AMM(자동 메모리 관리) 활성화

데이터베이스를 구성할 때 AMM(자동 메모리 관리) 기능을 활성화하지 않은 경우 다음 단계를 수행하여 활성화할 수 있습니다.

1. Database Home 페이지에서 Server 탭을 누릅니다.
2. Database Configuration 영역에서 Memory Advisors를 누릅니다.  
Memory Advisors 페이지가 나타납니다.
3. Automatic Memory Management에 대해 Enable을 누릅니다.  
Enable Automatic Memory Management 페이지가 나타납니다.
4. Automatic Memory Management에 대해 Total Memory Size와 Maximum Memory Size 값을 설정합니다.  
**주:** Maximum Memory Size를 변경하는 경우에는 데이터베이스 Instance를 재시작해야 합니다.
5. OK를 누릅니다.

Total Memory Size 필드 또는 MEMORY\_TARGET 초기화 파라미터의 값을 늘려서 나중에 크기를 늘릴 수도 있습니다. 그러나 이를 값을 Maximum Memory Size 필드 또는 MEMORY\_MAX\_TARGET 파라미터에서 지정하는 값보다 크게 설정할 수는 없습니다.

자세한 사항은 *Oracle Database Administrator's Guide*를 참조하십시오.

AMM을 활성화하고 나면 Memory Size Advisor를 사용하여 최대 및 대상 메모리 크기를 손쉽게 조정할 수 있습니다.

**주:** 오라클에서는 AMM(자동 메모리 관리)을 사용하여 메모리 관리 작업을 간소화할 것을 권장합니다.

# ASMM(자동 공유 메모리 관리) 활성화



The System Global Area (SGA) is a group of shared memory structures that contains data and control information for one Oracle database. The SGA is allocated in memory when an Oracle database instance is started.

Automatic Shared Memory Management **Disabled** **Enable** Enable을 눌러 자동 공유 메모리 관리를 활성화합니다.

Shared Pool	248 MB	Advice
Buffer Cache	136 MB	Advice
Large Pool	4 MB	
Java Pool	12 MB	
Other (MB)	5	
Total SGA (MB)	405	Calculate

**Maximum SGA Size**  
The Maximum SGA Size specifies the maximum memory that the database may allocate. If you specify the Maximum SGA Size, you can later dynamically change SGA component sizes (provided the total SGA size does not exceed the Maximum SGA Size).

Maximum SGA Size (MB)  The database must be restarted before any changes to this value take effect.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## ASMM(자동 공유 메모리 관리) 활성화

AMM을 활성화한 경우에는 ASMM(자동 공유 메모리 관리)이 자동으로 활성화됩니다. 데이터베이스를 구성할 때 AMM 또는 ASMM을 활성화하지 않은 경우 다음 단계를 수행하여 ASMM을 활성화할 수 있습니다.

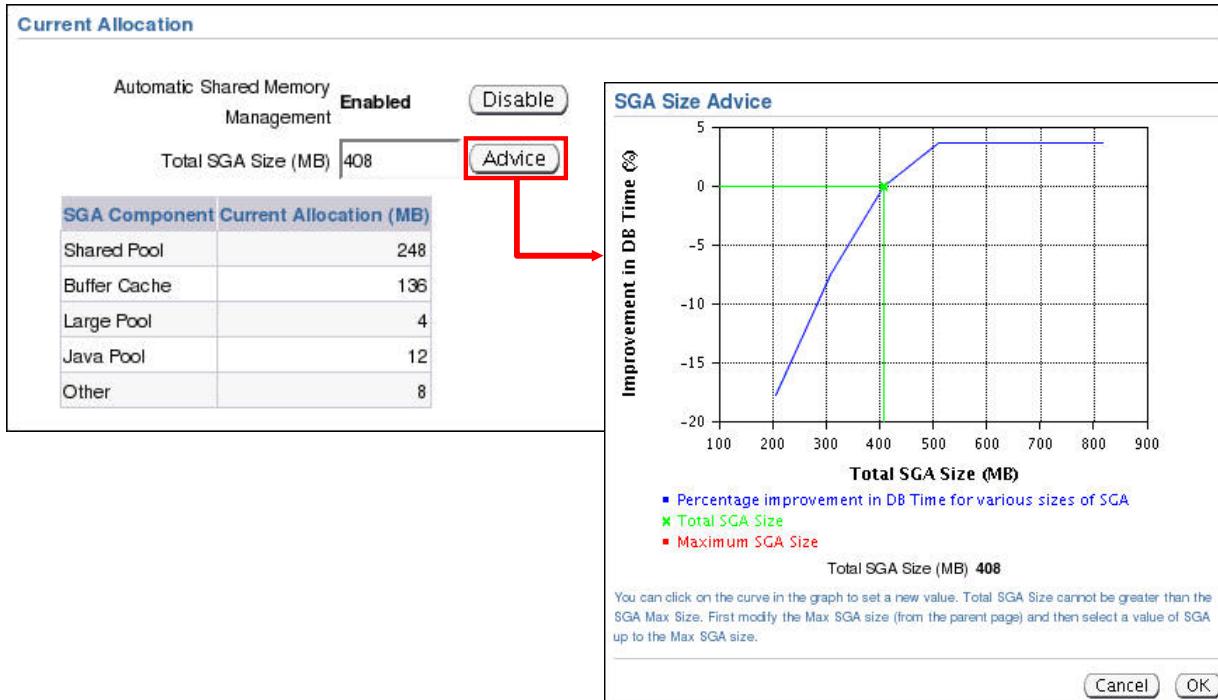
1. Database Home 페이지에서 Server 탭을 누릅니다.
2. Database Configuration 영역에서 Memory Advisors를 누릅니다.  
Memory Advisors 페이지가 나타납니다.
3. SGA 섹션 아래로 화면을 이동합니다. Automatic Shared Memory Management에서 Enable를 누릅니다.  
Enable Automatic Shared Memory Management 페이지가 나타납니다.
4. 총 SGA 크기를 지정합니다. OK를 누릅니다.

Total SGA Size 필드 또는 SGA\_TARGET 초기화 파라미터의 값을 늘려서 나중에 총 SGA 크기를 늘릴 수 있습니다. 그러나 이들 값을 Maximum SGA Size 필드 또는 SGA\_MAX\_SIZE 파라미터에서 지정하는 값보다 크게 설정할 수는 없습니다. 자세한 사항은 *Oracle Database Administrator's Guide*를 참조하십시오.

AMM을 비활성화하면 PGA Advisor에 액세스할 수 있습니다. PGA 메모리 값을 설정할 때는 PGA Advisor를 사용할 것을 권장합니다. PGA 탭을 눌러 PGA 속성 페이지에 액세스합니다. Advice를 눌러 PGA Advisor를 호출합니다.

**주:** 오라클에서는 ASMM(자동 공유 메모리 관리)을 사용하여 메모리 관리 작업을 간소화할 것을 권장합니다.

# Automatic Shared Memory Advisor



ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Automatic Shared Memory Advisor

ASMM이 활성화된 경우 ASMM이 관리하는 특정 공유 메모리 구성 요소에 대해 초기화 파라미터를 설정하지 않아야 합니다. ASMM이 활성화된 후에 SGA Size Advisor를 사용하여 총 SGA 크기에 대해 가장 적합한 값을 선택할 수 있습니다.

ASMM을 활성화하려는 경우 SPFILE의 개별 메모리 영역 파라미터를 설정할 경우 ASMM의 기능이 제한될 수 있으므로 이를 먼저 제거해야 합니다. ASMM 할당의 영향을 확인한 후에 특정 구성 요소 할당을 조정할지 결정하는 경우에는 해당 구성 요소에 대해 값을 지정할 수 있습니다. 지정한 값이 현재 값보다 작은 경우 이러한 값은 각 구성 요소의 최소 메모리 크기로 간주됩니다. 지정한 값이 현재 값보다 큰 경우 메모리 구성 요소의 크기는 사용 가능한 메모리가 있는 한 유저가 제공한 값에 맞게 최대한의 크기로 변경됩니다. 이렇게 하면 자동 조정에 사용할 수 있는 메모리 양이 제한되지만, 이 기능은 작업 환경에서 ASMM에서 수용하지 않는 특별한 크기 조정이 필요한 경우에 사용 가능합니다.

대상 초기화 파라미터는 다음과 같습니다.

- SHARED\_POOL\_SIZE
- LARGE\_POOL\_SIZE
- JAVA\_POOL\_SIZE
- DB\_CACHE\_SIZE
- STREAMS\_POOL\_SIZE

ASMM이 활성화 상태인 동안 이러한 파라미터를 조정하려면 ALTER SYSTEM 명령을 사용해야 합니다.

# Dynamic Performance 통계

시스템 차원	세션 특성	서비스 특성
<p>V\$SYSSTAT</p> <ul style="list-style-type: none"><li>• statistic#</li><li>• name</li><li>• class</li><li>• value</li><li>• stat_id</li></ul> <p>V\$SYSTEM_EVENT</p> <ul style="list-style-type: none"><li>• event</li><li>• total_waits</li><li>• total_timeouts</li><li>• time_waited</li><li>• average_wait</li><li>• time_waited_micro</li></ul> <p> 누적 통계  대기 이벤트</p>	<p>V\$SESSTAT</p> <ul style="list-style-type: none"><li>• sid</li><li>• statistic#</li><li>• value</li></ul> <p>V\$SESSION_EVENT</p> <ul style="list-style-type: none"><li>• sid</li><li>• event</li><li>• total_waits</li><li>• total_timeouts</li><li>• time_waited</li><li>• average_wait</li><li>• max_wait</li><li>• time_waited_micro</li><li>• event_id</li></ul>	<p>V\$SERVICE_STATS</p> <ul style="list-style-type: none"><li>• service_name_hash</li><li>• service_name</li><li>• stat_id</li><li>• stat_name</li><li>• value</li></ul> <p>V\$SERVICE_EVENT</p> <ul style="list-style-type: none"><li>• service_name</li><li>• service_name_hash</li><li>• event</li><li>• event_id</li><li>• total_waits</li><li>• total_timeouts</li><li>• time_waited</li><li>• average_wait</li><li>• time_waited_micro</li></ul>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## Dynamic Performance 통계

성능 문제를 효율적으로 진단하려면 통계를 사용할 수 있어야 합니다. Oracle 서버는 다양한 레벨의 세분성을 위해 여러 유형의 통계를 생성합니다.

시스템 레벨, 세션 레벨 및 서비스 레벨에서 대기 이벤트와 누적 통계가 모두 계산됩니다.

슬라이드에서 윗줄에는 누적 통계 뷰가 나와 있고 아랫줄에는 대기 이벤트 뷰가 나와 있습니다.

이러한 모든 범위에서 성능 문제를 분석하면 일반적으로 관심이 있는 기간 동안의 통계 변화(델타  $\Delta$ )를 확인할 수 있습니다. 가능한 대기 이벤트는 모두 V\$EVENT\_NAME 뷰에 카탈로그화됩니다. 모든 통계는 V\$STATNAME 뷰에 카탈로그화됩니다. 오라클 데이터베이스는 약 480개의 통계를 제공합니다.

## Dynamic Performance 통계(계속)

### 시스템 차원의 통계 표시

예제:

```
SQL> SELECT name, class, value FROM v$sysstat;
NAME                      CLASS      VALUE
-----
...
table scans (short tables)    64      135116
table scans (long tables)    64       250
table scans (rowid ranges)   64        0
table scans (cache partitions) 64        3
table scans (direct read)    64        0
table scan rows gotten      64     14789836
table scan blocks gotten     64      558542
...
```

시스템 차원의 통계는 튜닝 항목 및 디버깅 목적에 따라 분류됩니다. 이 분류에는 일반적인 Instance 작업, 리두 로그 베퍼 작업, Lock, 데이터베이스 베퍼 캐시 작업 등이 포함됩니다.

## 문제 해결 및 튜닝 뷰

### Instance/데이터베이스

V\$DATABASE  
V\$INSTANCE  
V\$PARAMETER  
V\$SPPARAMETER  
V\$SYSTEM\_PARAMETER  
V\$PROCESS  
V\$BGPPROCESS  
V\$PX\_PROCESS\_SYSSTAT  
V\$SYSTEM\_EVENT

### 디스크

V\$DATAFILE  
V\$FILESTAT  
V\$LOG  
V\$LOG\_HISTORY  
V\$DBFILE  
V\$TEMPFILE  
V\$TEMPSEG\_USAGE  
V\$SEGMENT\_STATISTICS

### 메모리

V\$BUFFER\_POOL\_STATISTICS  
V\$LIBRARYCACHE  
V\$SGAINFO  
V\$PGASTAT

### 경합

V\$LOCK  
V\$UNDOSTAT  
V\$WAITSTAT  
V\$LATCH

ORACLE

Copyright © 2010, Oracle. All rights reserved.

## 문제 해결 및 튜닝 뷰

슬라이드에는 성능 문제의 원인을 확인하거나 데이터베이스의 현재 상태를 분석하는 데 도움이 되는 일부 뷰가 나열되어 있습니다.

이러한 뷰에 대한 자세한 설명은 *Oracle Database Reference*를 참조하십시오.

## 부적합한 객체 및 사용할 수 없는 객체

### 성능에 미치는 영향:

- PL/SQL 코드 객체가 재컴파일됩니다.
- 인덱스가 재구축됩니다.



**ORACLE**

Copyright © 2010, Oracle. All rights reserved.

### 부적합한 객체 및 사용할 수 없는 객체

부적합한 PL/SQL 객체 및 사용할 수 없는 인덱스는 성능에 영향을 줍니다. 부적합한 PL/SQL 객체는 재컴파일해야 사용할 수 있습니다. 그러므로 PL/SQL 패키지, 프로시저 또는 함수에 액세스하려고 시도하는 첫번째 작업에 컴파일 시간이 추가됩니다. PL/SQL이 제대로 재컴파일되지 않으면 작업이 실패하고 오류가 발생합니다. 사용할 수 없는 인덱스는 옵티マイ저에서 무시됩니다. SQL 문의 적절한 성능이 사용할 수 없는 것으로 표시된 인덱스에 따라 달라지는 경우 인덱스를 재구축할 때까지는 성능이 개선되지 않습니다.

**부적합한 PL/SQL 객체:** 데이터 딕셔너리를 query하면 PL/SQL 객체의 현재 상태를 볼 수 있습니다. 다음을 사용하여 부적합한 PL/SQL 객체를 찾을 수 있습니다.

```
SELECT object_name, object_type FROM DBA_OBJECTS
WHERE status = 'INVALID';
```

기본적으로 Owner's Invalid Object Count metric은 24시간마다 확인됩니다. 개별 소유자에 대한 객체 수가 두 개를 초과하면 Alert가 발생됩니다.

INVALID상태의 PL/SQL 객체를 발견한 경우 먼저 이 객체가 이전에 VALID 상태였는지 여부를 알아야 합니다. 종종 응용 프로그램 개발자가 작동하지 않는 코드를 정리하지 않는 경우가 있습니다. PL/SQL 객체가 코드 오류로 인해 부적합한 상태인 경우에는 해당 오류를 수정하는 방법밖에 없습니다. 이전에 적합한 상태였던 프로시저가 최근에 부적합한 상태로 변경된 경우에는 다음과 같은 두 가지 방법으로 문제를 해결할 수 있습니다.

## 부적합한 객체 및 사용할 수 없는 객체(계속)

- 아무 작업도 하지 않습니다. 대부분의 PL/SQL 객체는 호출되었을 때 필요한 경우 자동으로 재컴파일됩니다. 객체를 재컴파일하는 동안 약간의 지연을 겪을 수 있으나 대부분 알아차릴 수 없을 만큼 짧은 시간입니다.
- 부적합한 객체를 수동으로 재컴파일합니다.

부적합한 PL/SQL 객체는 Enterprise Manager를 사용하거나 다음 SQL 명령을 통해 수동으로 재컴파일 할 수 있습니다.

```
ALTER PROCEDURE HR.add_job_history COMPILE;
```

**PL/SQL 패키지를** 수동으로 재컴파일하려면 다음 두 단계가 필요합니다.

```
ALTER PACKAGE HR.maintainemp COMPILE;
```

```
ALTER PACKAGE HR.maintainemp COMPILE BODY;
```

**사용할 수 없는 인덱스:** 부적합한 인덱스는 DBA\_INDEXES 데이터 딕셔너리 뷰를 query하여 찾을 수 있습니다.

```
SELECT index_name, table_name FROM DBA_INDEXES
WHERE status = 'UNUSABLE';
```

Partition 인덱스의 경우 상태는 DBA\_IND\_PARTITIONS 뷰에 보관됩니다.

사용할 수 없는 인덱스는 포인터를 다시 계산하여 재구축하면 적합한 상태가 됩니다. 사용할 수 없는 인덱스를 재구축하면 새 위치에 인덱스가 재생성된 다음 사용할 수 없는 인덱스가 삭제됩니다. 이는 Enterprise Manager를 사용하거나 다음 SQL 명령을 통해 수행할 수 있습니다.

```
ALTER INDEX HR.emp_empid_pk REBUILD;
```

```
ALTER INDEX HR.emp_empid_pk REBUILD ONLINE;
```

```
ALTER INDEX HR.email REBUILD TABLESPACE USERS;
```

TABLESPACE 절을 생략하면 인덱스가 존재하는 동일한 테이블스페이스에 재구축됩니다.

REBUILD ONLINE 절을 사용하면 재구축하는 동안 유저가 인덱스의 테이블을 계속 갱신할 수 있습니다. (ONLINE 키워드가 없는 경우 영향을 받는 테이블에 대해 DML을 수행하려면 재구축이 완료될 때까지 기다려야 합니다. 사용할 수 없는 상태인 인덱스는 ONLINE 키워드를 사용하는 경우에도 재구축 중에 사용되지 않습니다.)

Enterprise Manager는 Reorganize 작업을 사용하여 UNUSABLE 인덱스를 재구성합니다.

**주:** 인덱스를 재구축하려면 재구축에 필요한 사용 가능한 공간이 있어야 합니다. 재구축 작업을 시작하기 전에 공간이 충분한지 확인하십시오. Enterprise Manager는 필요한 공간을 자동으로 검사합니다.

## 퀴즈

자동 메모리 관리를 사용하면 Oracle Instance가 \_\_\_\_\_에서 SGA로 메모리를 재할당할 수 있습니다.

1. Large Pool
2. 로그 버퍼
3. PGA
4. Streams Pool

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 3

## 퀴즈

**SGA\_TARGET은 \_\_\_\_\_보다 크게 설정할 수 없습니다.**

- 1. LOG\_BUFFER**
- 2. SGA\_MAX\_SIZE**
- 3. STREAMS\_POOL\_SIZE**
- 4. PGA\_AGGREGATE\_TARGET**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

## 요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- Enterprise Manager를 사용하여 성능 모니터
- AMM(자동 메모리 관리) 사용
- Memory Advisor를 사용하여 메모리 버퍼의 크기 조정
- 성능 관련 동적 뷰 보기
- invalid 및 unusable 객체 문제 해결



Copyright © 2010, Oracle. All rights reserved.

## 연습 13 개요: 성능 모니터 및 개선

이 연습에서는 다음 내용을 다룹니다.

- 사용할 수 없는 인덱스 감지 및 복구
- Enterprise Manager에서 Performance 페이지 사용



Copyright © 2010, Oracle. All rights reserved.

