

Oracle Database 11g: Administration Workshop I(한글판)

볼륨 II · 학생용

D50102KR20

Edition 2.0

2010년 5월

D67249

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

셀 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

14

백업 및 Recovery 개념

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 오라클 데이터베이스에서 발생할 수 있는 Failure 유형 식별
- instance recovery 튜닝 방법 설명
- 체크포인트, 리두 로그 파일 및 아카이브 로그 파일의 중요성 파악
- Fast Recovery Area 구성
- ARCHIVELOG 모드 구성

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

관리자의 임무

데이터베이스 관리자의 임무는 다음과 같습니다.

- 어느 곳에서나 발생할 수 있는 failure로부터 데이터베이스 보호
- MTBF(Mean time between failure) 늘리기
- 중복을 통한 중요 구성 요소 보호
- MTTR(Mean time to recover) 줄이기
- 데이터 손실 최소화

ORACLE

Copyright © 2010, Oracle. All rights reserved.

관리자의 임무

데이터베이스 관리자(DBA)의 목표는 유저가 필요로 할 때 데이터베이스가 열려 있고 사용 가능하도록 하는 것입니다. 이러한 목표를 달성하기 위해 DBA는 대개 시스템 관리자와 함께 다음 작업을 수행합니다.

- Failure의 일반적인 원인을 예상하고 예방 대책을 마련합니다.
- 가용성을 저하시키는 MTBF(Mean time between failure)를 늘리기 위한 작업을 수행합니다.
- 하드웨어가 최대한 안정적인 상태이고, 중복을 통해 주요 구성 요소를 보호하며, 운영 체제 유지 관리를 적시에 수행하고 있는지 확인합니다. 오라클 데이터베이스는 MTBF를 증가시키기 위한 다음과 같은 고급 구성 옵션을 제공합니다.
 - Real Application Clusters(*Oracle Database 11g: Real Application Clusters* 과정에서 설명)
 - Streams(*Oracle Database 11g: Implement Streams* 과정에서 설명)
 - Oracle Data Guard(*Oracle Database 11g: Data Guard Administration* 과정에서 설명)
- 미리 Recovery 프로시저를 연습하고 백업을 구성하여 필요할 때 즉시 사용 가능하도록 함으로써 MTTR(Mean time to recover)을 감소시킵니다.
- 데이터 손실을 최소화합니다. 승인된 최적의 사용법을 따르는 DBA는 커밋된 트랜잭션이 손실되지 않도록 데이터베이스를 구성할 수 있습니다. 이러한 기능을 보장하기 위해 다음 항목이 지원됩니다.
 - 아카이브 로그 파일(이 단원의 뒷부분에서 설명)
 - Flashback 기술
 - Standby Database 및 Oracle Data Guard(*Oracle Database 11g: Data Guard Administration* 과정에서 설명)

Failure의 범주

Failure는 일반적으로 다음 범주로 나눌 수 있습니다.

- **Statement Failure**
- **User Process Failure**
- **Network Failure**
- **User error**
- **Instance Failure**
- **Media Failure**



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Failure의 범주

- **Statement Failure:** 단일 데이터베이스 작업(선택, 삽입, 갱신 또는 삭제)이 실패하는 경우
- **User process failure:** 단일 데이터베이스 세션이 실패하는 경우
- **Network failure:** 데이터베이스 연결이 끊깁니다.
- **User error:** 유저가 성공적으로 작업을 완료했지만 작업이 정확하지 않습니다(테이블 삭제 또는 부정확한 데이터 입력).
- **Instance failure:** 데이터베이스 instance가 예상치 않게 종료됩니다.
- **Media failure:** 데이터베이스 작업에 필요한 파일이 없습니다(즉, 파일이 삭제되었거나 디스크가 실패함).

Statement Failure

일반적인 문제	가능한 해결 방법
부적합한 데이터를 테이블에 입력하려고 시도한 경우	유저와 함께 데이터를 검증하고 수정합니다.
권한이 부족한 상태에서 작업을 수행하려고 시도한 경우	적절한 객체나 시스템 권한을 제공합니다.
공간 할당 시도에 실패한 경우	<ul style="list-style-type: none">Resumable Space Allocation 활성화소유자 할당량을 늘립니다.테이블스페이스에 공간을 추가합니다.
응용 프로그램에 논리 오류가 발생한 경우	개발자와 함께 프로그램 오류를 수정합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Statement Failure

단일 데이터베이스 작업을 실패한 경우 유저 권한이나 데이터베이스 공간 할당으로 인한 오류를 수정하려면 DBA의 개입이 필요할 수 있습니다. DBA는 또한 자신의 작업 영역에 직접 속하지 않는 문제라도 해결하는 데 도움을 제공해야 할 수 있습니다. 이는 조직마다 크게 다를 수 있습니다. 예를 들어, 기성품 응용 프로그램을 사용하는 조직(소프트웨어 개발자가 없는 조직)의 경우 DBA가 제품에 대한 문의를 모두 처리하고 응용 프로그램의 논리 오류를 조사해야 합니다.

응용 프로그램의 논리 오류를 이해하기 위해서는 개발자와 협력하여 문제의 범위를 확인해야 합니다. 오라클 데이터베이스 도구는 Audit Trail 또는 이전 트랜잭션 검사를 도와 줄 수 있습니다.

주: 여러 경우에 Statement Failure는 의도적인 것이며 필요에 의한 것입니다. 예를 들어, 보안 정책 및 할당량 규칙은 미리 결정되어지는 경우가 많습니다. 유저가 자신의 한도를 초과하여 오류가 발생한 경우, 작업이 실패하도록 두는 것이 바람직할 수 있으며, 해결 방법은 필요하지 않을 수 있습니다.

User Process Failure

일반적인 문제	가능한 해결 방법
유저가 비정상적으로 연결을 끊은 경우	User Process Failure를 해결하는 데는 대개 DBA 작업이 필요하지 않습니다. instance 백그라운드 프로세스가 커밋되지 않은 변경 사항을 롤백하고 lock을 해제합니다.
유저 세션이 비정상적으로 종료되었습니다.	
세션을 종료시키는 프로그램 오류가 발생했습니다.	경향을 살펴봅니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

User Process Failure

비정상적으로 instance 연결을 끊은 User Process에서는 롤백이 필요한 커밋되지 않은 작업이 진행 중일 수 있습니다. PMON(Process Monitor) 백그라운드 프로세스는 해당 세션이 계속 연결되어 있는지 확인하기 위해 주기적으로 서버 프로세스를 폴링합니다. PMON에서 서버 프로세스에 더 이상 유저가 연결되지 않은 것으로 나타나면 PMON은 진행 중인 모든 트랜잭션을 Recovery합니다. 또한 커밋되지 않은 변경 사항을 롤백하고 실패한 세션이 보유한 Lock을 해제합니다.

User Process Failure로부터 Recovery하는 데는 DBA의 개입이 필요하지 않지만 관리자는 경향을 주시해야 합니다. 비정상적으로 연결이 끊긴 한두 명의 유저는 문제가 되지 않습니다. 낮은 비율의 User Process Failure는 이따금 발생할 수 있습니다.

그러나 시스템과 관련된 Failure가 일관되게 발생하면 다른 문제가 있는 것입니다. 비정상적인 연결 끊김 비율이 높은 경우에는 프로그램을 단순히 종료하기보다는 로그아웃하도록 알려주는 등 유저에 대한 교육이 필요합니다. 또한 네트워크나 응용 프로그램에 문제가 있는 것일 수 있습니다.

Network Failure

일반적인 문제	가능한 해결 방법
리스너가 실패한 경우	백업 리스너와 Connect-Time Failover를 구성합니다.
NIC(네트워크 인터페이스 카드)가 실패한 경우	다중 네트워크 카드를 구성합니다.
네트워크 연결이 실패한 경우	백업 네트워크 연결을 구성합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Network Failure

Network Failure의 가장 좋은 해결 방법은 네트워크 연결을 위한 중복 경로를 제공하는 것입니다. 백업 리스너, 네트워크 연결 및 네트워크 인터페이스 카드는 시스템 가용성에 영향을 주는 Network Failure가 발생할 가능성을 줄여줍니다.

User Error

일반적인 원인	가능한 해결 방법
유저가 실수로 데이터를 삭제하거나 수정한 경우	트랜잭션 및 종속 트랜잭션을 롤백하거나 테이블을 되돌립니다.
유저가 테이블을 삭제한 경우	Recycle bin에서 테이블을 Recovery합니다.



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

User Error

유저가 실수로 데이터를 삭제하거나 수정할 수 있습니다. 해당 작업을 아직 커밋하지 않았거나 프로그램을 종료하지 않은 경우에는 롤백을 수행하면 됩니다.

Oracle LogMiner에서는 Enterprise Manager 또는 SQL 인터페이스를 통해 온라인 리두 로그와 아카이브된 리두 로그를 query할 수 있습니다. 트랜잭션 데이터는 언두 세그먼트에서보다 온라인 리두 로그에서 더 오래 지속될 수 있으며 리두 정보를 아카이브하도록 구성한 경우 아카이브된 파일을 삭제할 때까지 리두가 지속됩니다. Oracle LogMiner에 대해서는 *Oracle Database Utilities* 참조 설명서에서 설명합니다.

유저가 테이블을 삭제한 경우 삭제 전 상태로 테이블을 Flashback하면 Recycle bin에서 테이블을 Recovery할 수 있습니다. Flashback 기술에 대해서는 *Oracle Database 11g: Administration Workshop II* 과정에서 자세하게 설명합니다.

Recycle bin이 이미 지워졌거나 유저가 PURGE 옵션으로 테이블을 삭제한 경우에도 데이터베이스가 올바르게 구성되어 있다면 PITR(Point-In-Time Recovery)을 사용하여 삭제된 테이블을 여전히 Recovery할 수 있습니다. PITR은 *Oracle Database 11g: Administration Workshop II* 과정과 *Oracle Database Backup and Recovery User's Guide*에서 설명합니다.

Flashback 기술

Flashback 기술을 사용하여 다음을 수행합니다.

- 데이터의 과거 상태 보기
- 데이터의 시점을 앞뒤로 이동
- 유저의 오류 분석 및 Recovery 지원



오류 분석용 도구

Oracle Flashback Query

Oracle Flashback Versions Query

Oracle Flashback Transaction Query

오류 Recovery용 도구

Oracle Flashback Transaction Backout

Oracle Flashback Table

Oracle Flashback Drop

Oracle Flashback Database

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Flashback 기술

오라클 데이터베이스는 데이터베이스를 백업에서 복원할 필요 없이 데이터의 과거 상태를 보고 시점을 앞뒤로 이동할 수 있는 기능 그룹인 Oracle Flashback 기술을 제공합니다. 이 기술을 사용하면 유저가 오류를 분석하고 Recovery할 수 있습니다. 잘못된 변경 내용을 커밋한 유저의 경우 다음을 사용하여 오류를 분석합니다.

- Flashback Query:** 커밋된 데이터를 과거의 특정 시점에 존재했던 상태로 봅니다.
SELECT 명령을 AS OF 절과 함께 사용하면 시간 기록 또는 SCN을 통해 과거의 시점을 참조할 수 있습니다.
- Flashback Version Query:** 특정 시간 간격 동안 커밋된 기록 데이터를 봅니다. 기존 인덱스의 성능을 보존하기 위해 SELECT 명령의 VERSIONS BETWEEN 절을 사용합니다.
- Flashback Transaction Query:** 트랜잭션 레벨에서 수행된 모든 데이터베이스 변경 내용을 봅니다.

User Error를 Recovery하는 데 사용할 수 있는 방법은 다음과 같습니다.

- Flashback Transaction Backout:** 특정 트랜잭션 및 종속 트랜잭션을 롤백합니다.
- Flashback Table:** 다른 데이터베이스 객체에 영향을 주지 않고 하나 이상의 테이블을 이전 시점의 컨텐트로 되돌립니다.
- Flashback Drop:** 삭제된 테이블을 인덱스 및 트리거 등의 종속 객체와 함께 Recycle bin에서 데이터베이스로 되돌림으로써 테이블 삭제 작업을 취소합니다.
- Flashback Database:** 데이터베이스를 과거 시점 또는 SCN(시스템 변경 번호)로 되돌립니다.

Instance Failure

일반적인 원인	가능한 해결 방법
정전	STARTUP 명령을 사용하여 Instance를 재시작합니다. Instance Failure의 Recovery는 자동으로 이루어집니다. 리두 로그의 변경 사항이 롤포워드되고 커밋되지 않은 트랜잭션이 롤백됩니다.
하드웨어 Failure	Alert log, Trace file 및 Enterprise Manager를 사용하여 failure의 원인을 조사합니다.
Critical 백그라운드 프로세스 중 하나의 Failure	Alert log, Trace file 및 Enterprise Manager를 사용하여 failure의 원인을 조사합니다.
비상 종료 프로시저	Alert log, Trace file 및 Enterprise Manager를 사용하여 failure의 원인을 조사합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Instance Failure

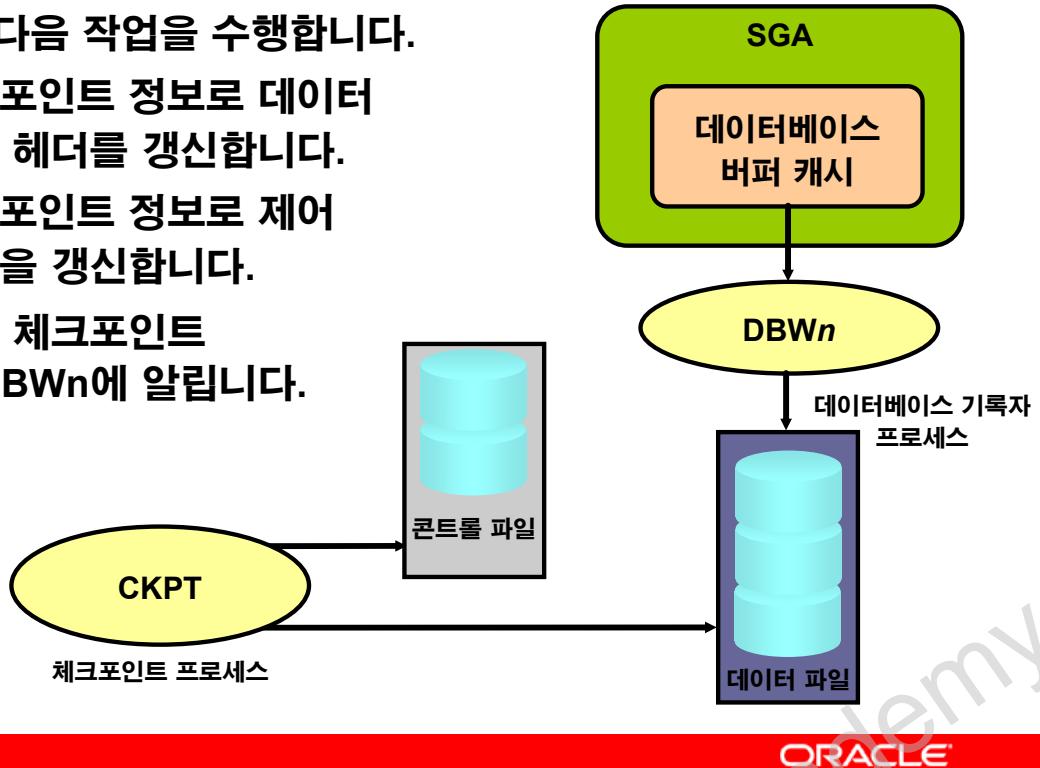
모든 데이터베이스 파일을 동기화하기 전에 데이터베이스 instance가 종료된 경우 Instance failure가 발생합니다. Instance failure는 하드웨어/소프트웨어 failure로 인해 발생하거나 비상 종료 명령인 SHUTDOWN ABORT 및 STARTUP FORCE를 사용한 경우 발생할 수 있습니다.

Oracle Restart가 활성화되었고 데이터베이스를 모니터 중인 경우 Instance Failure에서 Recovery를 수행할 때 관리자가 관여할 필요가 거의 없습니다. Oracle Restart는 실패하는 즉시 데이터베이스 Instance를 재시작하려고 시도합니다. 수동 간섭이 필요한 경우에는 메모리 CPU Failure와 같은 Instance의 재시작을 방해하는 보다 심각한 문제가 있을 수 있습니다.

Instance Recovery 이해: CKPT(체크포인트) 프로세스

CKPT는 다음 작업을 수행합니다.

- 체크포인트 정보로 데이터 파일 헤더를 갱신합니다.
- 체크포인트 정보로 제어 파일을 갱신합니다.
- 전체 체크포인트 시 DBWn에 알립니다.



Copyright © 2010, Oracle. All rights reserved.

Instance Recovery 이해: CKPT(체크포인트) 프로세스

Instance Recovery를 이해하려면 특정 백그라운드 프로세스의 작동 원리를 이해해야 합니다.

3초마다(또는 이보다 자주), CKPT 프로세스는 DBWn이 SGA의 수정된 데이터 블록을 디스크에 기록한 내용을 콘트롤 파일에 저장합니다. 이를 "incremental 체크포인트"라고 합니다. 체크포인트의 목적은 온라인 리두 로그 파일에서 instance recovery가 시작될 위치("체크포인트 위치"라고 함)를 식별하는 것입니다.

로그 스위치가 발생할 경우 CKPT 프로세스는 이 체크포인트 정보를 데이터 파일의 헤더에 기록합니다.

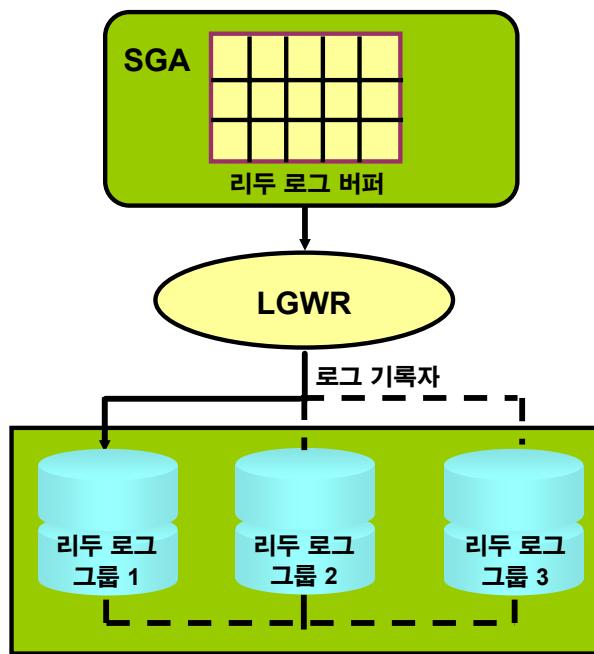
체크포인트는 다음과 같은 이유로 존재합니다.

- 메모리에 있는 수정된 데이터 블록을 정기적으로 디스크에 기록하므로 시스템이나 데이터베이스에 Failure가 발생할 경우 데이터가 손실되지 않습니다.
- Instance Recovery에 필요한 시간을 줄입니다. Recovery를 수행하기 위해서는 마지막 체크포인트 이후의 온라인 리두 로그 파일 항목만 처리되면 됩니다.
- 종료하는 동안 모든 커밋된 데이터가 데이터 파일에 기록됨을 보장합니다.

CKPT 프로세스에 의해 기록된 체크포인트 정보에는 체크포인트 위치, SCN(시스템 변경 번호), Recovery를 시작할 온라인 리두 로그 파일의 위치, 로그에 대한 정보 등이 포함됩니다.

주: CKPT 프로세스는 데이터 블록을 디스크에 기록하거나 리두 블록을 온라인 리두 로그 파일에 기록하지 않습니다.

Instance Recovery 이해: 리두 로그 파일 및 로그 기록자



리두 로그 파일의 특징은 다음과 같습니다.

- 데이터베이스 변경 사항을 기록합니다.
- 데이터 손실로부터 보호하려면 다중화되어야 합니다.

로그 기록자는 다음과 같은 경우에 데이터를 기록합니다.

- 커밋 시
- 1/30이 찼을 때
- 3초마다
- DBWn 쓰기 전
- 정상 종료 전

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Instance Recovery 이해: 리두 로그 파일 및 로그 기록자

리두 로그 파일에는 트랜잭션 및 내부 Oracle 서버 작업의 결과로 발생한 데이터베이스 변경 사항이 기록됩니다. (트랜잭션은 논리적 작업 단위로, 유저가 실행한 하나 이상의 SQL 문으로 구성됩니다.) 리두 로그 파일은 정전, 디스크 Failure 등의 이유로 시스템 Failure가 발생하는 경우 데이터베이스 무결성이 손실되지 않도록 합니다. 디스크 failure 발생 시 리두 로그 파일에 저장된 정보가 손실되지 않도록 하려면 파일을 다중화해야 합니다.

리두 로그는 여러 그룹의 리두 로그 파일들로 구성됩니다. 단일 그룹은 하나의 리두 로그 파일과 이를 다중화한 복사본으로 구성됩니다. 동일한 각각의 복사본은 해당 그룹의 멤버이며 각 그룹은 숫자로 식별됩니다. LGWR(로그 기록자) 프로세스는 파일이 가득 차거나 로그 스위치 작업이 요청될 때까지 리두 로그 버퍼의 리두 레코드를 리두 로그 그룹의 모든 멤버에 기록합니다. 그런 후에 다음 그룹의 파일로 전환하여 계속 기록합니다. 리두 로그 그룹은 순환 방식으로 사용됩니다.

최적의 사용법(Best Practice Tip): 가능하면 다중화된 리두 로그 파일은 서로 다른 디스크에 있어야 합니다.

Instance Recovery 이해

자동 Instance Recovery 또는 Crash Recovery의 특징은 다음과 같습니다.

- 종료 시 동기화되지 않은 파일이 있는 데이터베이스를 열려고 할 때 발생합니다.
- 리두 로그 그룹에 저장된 정보를 사용하여 파일을 동기화합니다.
- 두 개의 개별 작업을 포함합니다.
 - 롤포워드: 리두 로그 변경 사항(커밋된 사항 및 커밋되지 않은 내용 모두)은 데이터 파일에 적용됩니다.
 - 롤백: 변경되었지만 커밋되지 않은 내용이 원래 상태로 돌아갑니다.

ORACLE®

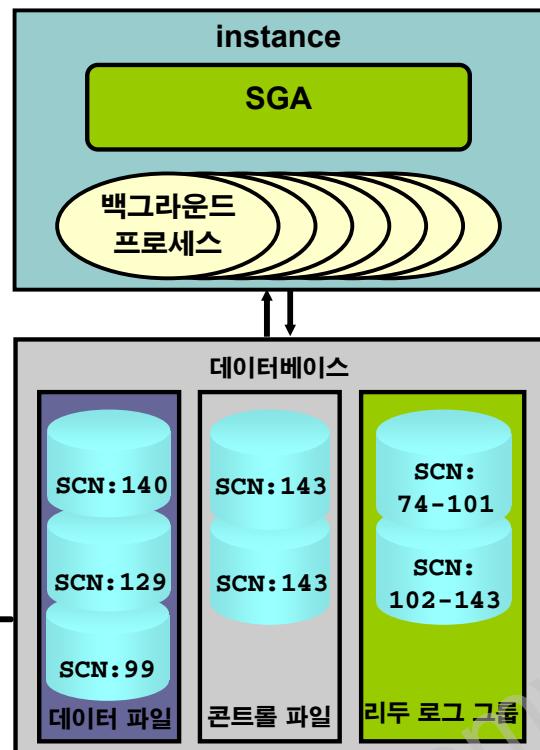
Copyright © 2010, Oracle. All rights reserved.

Instance Recovery

오라클 데이터베이스는 자동으로 Instance Failure로부터 Recovery합니다. 이 경우에는 Instance를 정상적으로 시작하기만 하면 됩니다. Oracle Restart가 활성화되어 있고 이 데이터베이스를 모니터하도록 구성된 경우 이 작업이 자동으로 수행됩니다. Instance가 콘트롤 파일을 마운트한 다음 데이터 파일을 열려고 시도합니다. Instance는 종료 시에 데이터 파일이 동기화되지 않았음을 발견하면 리두 로그 그룹에 포함된 정보를 사용하여 데이터 파일을 종료 시점으로 롤포워드합니다. 그런 다음 데이터베이스가 열리고 커밋되지 않은 트랜잭션이 모두 롤백됩니다.

Instance Recovery의 단계

1. Instance 시작(데이터 파일이 동기화되지 않음)
2. 롤포워드(리두)
3. 파일에 커밋된 데이터와 커밋되지 않은 데이터 포함
4. 데이터베이스가 열림
5. 롤백(언두)
6. 파일에 커밋된 데이터 포함



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Instance Recovery의 단계

Instance가 데이터 파일을 열려면 데이터 파일 헤더에 있는 SCN(시스템 변경 번호)이 데이터베이스 콘트롤 파일에 저장된 현재 SCN과 일치해야 합니다.

번호가 일치하지 않을 경우 instance는 온라인 리두 로그로부터 리두 데이터를 적용하고 데이터 파일이 최신 상태가 될 때까지 순차적으로 트랜잭션을 "리두"합니다. 모든 데이터 파일이 콘트롤 파일과 동기화되면 데이터베이스가 열리고 유저가 로그인할 수 있습니다.

리두 로그가 적용될 때 데이터베이스가 Failure 당시의 상태가 되도록 모든 트랜잭션이 적용됩니다. 여기에는 대개 진행 중이었지만 아직 커밋되지 않은 트랜잭션이 포함됩니다. 데이터베이스가 열린 후에는 이러한 커밋되지 않은 트랜잭션이 롤백됩니다. Instance Recovery의 롤백 단계를 마치면 데이터 파일에는 커밋된 데이터만 포함됩니다.

Instance Recovery 튜닝

- Instance Recovery 동안 체크포인트 위치와 리두 로그 끝 사이의 트랜잭션이 데이터 파일에 적용되어야 합니다.
- 체크포인트 위치와 리두 로그 끝 사이의 차이를 제어함으로써 Instance Recovery를 튜닝합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Instance Recovery 튜닝

Instance가 트랜잭션에 대해 commit complete를 반환하기 전에 트랜잭션 정보는 리두 로그 그룹에 기록됩니다. 리두 로그 그룹에 있는 정보는 Failure가 발생한 경우 트랜잭션이 Recovery될 수 있도록 보장합니다. 또한 트랜잭션 정보는 데이터 파일에 기록되어야 합니다. 데이터 파일 쓰기 프로세스는 리두 쓰기보다 훨씬 느리기 때문에 데이터 파일 쓰기는 대개 리두 로그 그룹에 정보가 기록된 후에 발생합니다. (데이터 파일에 대한 임의의 쓰기는 리두 로그 파일에 대한 순차 쓰기보다 느립니다.)

체크포인트 프로세스는 3초마다 리두 로그의 체크포인트 위치에 대한 정보를 콘트롤 파일에 기록합니다. 따라서 이 시점 이전에 기록된 일부 리두 로그 항목은 데이터베이스 Recovery에 필요하지 않습니다. 슬라이드의 그림에서 줄무늬로 표시된 블록이 아직 디스크에 기록되지 않았습니다.

Instance Recovery에 필요한 시간은 데이터 파일을 마지막 체크포인트에서 콘트롤 파일에 기록된 최근 SCN으로 가져오는 데 필요한 시간입니다. 관리자는 MTTR 목표(초 단위)를 설정하고 리두 로그 그룹의 크기를 지정하여 이 시간을 제어합니다. 예를 들어, 두 리두 그룹에서 체크포인트 위치와 리두 로그 그룹 끝 사이의 거리는 가장 작은 리두 로그 그룹의 90%보다 클 수 없습니다.

MTTR Advisor 사용

- 초 또는 분 단위로 원하는 시간을 지정합니다.
- 기본값은 0입니다(비활성화됨).
- 최대값은 3,600초(1시간)입니다.

The screenshot shows the Oracle Enterprise Manager interface for a database instance named 'orcl.oracle.com'. The 'Availability' tab is selected. Under the 'Backup/Recovery' section, the 'Recovery Settings' tab is highlighted with a red box and a red arrow pointing from the previous slide's note to this tab. The 'Recovery Settings' page displays the 'Instance Recovery' configuration, which includes a text area explaining the fast-start checkpointing feature and a configuration section for the 'Desired Mean Time To Recover' (set to 15 seconds). Buttons for 'Show SQL', 'Revert', and 'Apply' are visible at the top right of this section.

MTTR Advisor 사용

MTTR 목표 설정과 관련한 지원을 보려면 다음 중 하나를 선택하십시오.

- Enterprise Manager > Advisor Central(Related Links 섹션) > MTTR Advisor
- Enterprise Manager > Availability > Recovery Settings

FAST_START_MTTR_TARGET 초기화 파라미터는 Instance 또는 시스템 Failure에서 Recovery 시간 구성을 단순화합니다. MTTR Advisor는 원하는 시간이나 가능한 근접한 시간에 Instance Recovery를 활성화할 수 있도록 FAST_START_MTTR_TARGET 값을 여러 파라미터로 변환합니다. FAST_START_MTTR_TARGET 파라미터를 명시적으로 0으로 설정하면 MTTR Advisor가 비활성화됩니다.

FAST_START_MTTR_TARGET 파라미터는 시스템에 대한 서비스 레벨 합의를 지원하는 값으로 설정해야 합니다. MTTR 목표를 작은 값으로 설정하면 추가적인 데이터 파일 쓰기로 인해 I/O 오버헤드가 증가하므로 성능에 영향을 줍니다. 그러나 MTTR 목표를 너무 크게 설정하면 Instance에서 Crash 후 Recovery에 걸리는 시간이 늘어납니다.

Media Failure

일반적인 원인	가능한 해결 방법
디스크 드라이브 Failure가 발생한 경우	1. 백업으로부터 손상된 파일을 복원합니다. 2. 필요한 경우 새 파일의 위치를 데이터베이스에 알립니다.
디스크 컨트롤러 Failure가 발생한 경우	3. 필요한 경우 리두 정보를 적용하여 파일을 Recovery합니다.
데이터베이스 작업에 필요한 파일 삭제 또는 손상	

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Media Failure

Oracle은 *media failure*를 하나 이상의 데이터베이스 파일(데이터, 콘트롤 또는 리두 로그 파일)을 손실하거나 손상시키는 failure로 정의합니다.

Media Failure를 Recovery하려면 누락된 파일을 복원하고 Recovery해야 합니다. 데이터베이스가 Media Failure로부터 Recovery될 수 있도록 하려면 다음 페이지에서 설명하는 최적의 사용법을 따르십시오.

Recovery 가능성 구성

Recovery 가능성을 최대화하도록 데이터베이스를 구성하려면 다음과 같이 하십시오.

- 정기적 백업의 일정을 잡습니다.
- 콘트롤 파일을 다중화합니다.
- 리두 로그 그룹을 다중화합니다.
- 리두 로그의 아카이브된 복사본을 보유합니다.

The screenshot shows the Oracle Database Enterprise Manager interface. The top navigation bar includes Home, Performance, Availability (which is selected and highlighted in blue), Server, Schema, Data Movement, and Software and Support. Below the navigation bar, there are two main sections: 'Backup/Recovery' and 'Oracle Secure Backup'. The 'Backup/Recovery' section contains links for Setup (Backup Settings, Recovery Settings, Recovery Catalog Settings) and Manage (Schedule Backup, Manage Current Backups, Backup Reports, Manage Restore Points, Perform Recovery, View and Manage Transactions). The 'Oracle Secure Backup' section contains links for Oracle Secure Backup Device and Media, and File System Backup and Restore. At the bottom of the interface, there is a red banner with the ORACLE logo and a copyright notice: Copyright © 2010, Oracle. All rights reserved.

Recovery 가능성 구성

데이터를 최대한 보호하려면 다음을 수행해야 합니다.

- 정기적 백업 일정 잡기
대부분의 Media Failure에서는 손실되거나 손상된 파일을 백업에서 복원해야 합니다.
- 콘트롤 파일 다중화
데이터베이스와 연관된 모든 콘트롤 파일은 동일합니다. 단일 콘트롤 파일이 손실된 경우에는 어렵지 않게 Recovery할 수 있지만 모든 콘트롤 파일이 손실된 경우에는 Recovery하기가 훨씬 어렵습니다. 최소한 두 개의 복사본을 보유하여 모든 콘트롤 파일이 손실되지 않도록 보호합니다.
- 리두 로그 그룹 다중화
Instance failure 또는 Media failure로부터 recovery하려면 리두 로그 정보를 사용하여 데이터 파일을 마지막 커밋된 트랜잭션으로 롤포워드합니다. 리두 로그 그룹이 단일 리두 로그 파일에 의존하는 경우 해당 파일이 손실되면 데이터가 손실될 수 있습니다. 그러므로 각 리두 로그 그룹에 대해 최소한 두 개의 복사본을 보관하고, 가능하면 각 복사본을 서로 다른 디스크 컨트롤러에 보관합니다.
- 리두 로그의 아카이브된 복사본 보유
파일이 손실된 후 백업을 통해 복원되는 경우 Instance는 리두 정보를 적용하여 해당 파일을 콘트롤 파일에 포함된 최근 SCN으로 가져와야 합니다. 기본 설정을 사용할 경우 데이터 파일에 리두 정보가 기록된 후 데이터베이스가 해당 정보를 겹쳐쓸 수 있습니다. 리두 로그의 아카이브된 복사본에 리두 정보를 보유하도록 데이터베이스를 구성할 수 있습니다. 이렇게 하려면 데이터베이스를 ARCHIVELOG 모드에 두면 됩니다.

구성 작업은 Enterprise Manager 또는 명령행을 사용하여 수행할 수 있습니다.

Fast Recovery Area 구성

Fast Recovery Area:

- 간편한 백업 저장 영역 관리를 위해 강력하게 권장됨
- 작업 중인 데이터베이스 파일과는 구분되는 저장 공간
- DB_RECOVERY_FILE_DEST 파라미터로 지정된 위치
- DB_RECOVERY_FILE_DEST_SIZE 파라미터로 지정된 크기
- 백업, 아카이브된 로그, Flashback 로그, 다중화된 콘트롤 파일 및 다중화된 리두 로그를 모두 보관할 수 있을 만큼 충분히 큼
- Retention 정책에 따라 자동으로 관리됨

Fast Recovery Area를 구성하면 위치, 크기 및 Retention 정책이 결정됩니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Fast Recovery Area 구성

Fast Recovery Area는 아카이브된 로그, 백업, Flashback 로그, 다중화된 콘트롤 파일 및 다중화된 리두 로그를 보관하기 위해 디스크에서 비워 두는 공간입니다. Fast Recovery Area를 사용하면 백업 저장 영역 관리가 간편해지므로 사용할 것을 적극 권장합니다. Fast Recovery Area는 데이터베이스 데이터 파일과 기본 온라인 로그 파일 및 콘트롤 파일의 위치와 구분된 저장 공간에 배치해야 합니다.

Fast Recovery Area에 할당할 디스크 공간의 양은 데이터베이스의 크기 및 작업 레벨에 따라 달라집니다. 일반적으로는 Fast Recovery Area가 클수록 보다 유용하게 활용할 수 있습니다. Fast Recovery Area는 데이터 및 콘트롤 파일 복사본을 비롯하여 Retention 정책을 기반으로 보관된 백업과 함께 데이터베이스를 Recovery하는 데 필요한 Flashback 로그, 온라인 리두 로그 및 아카이브된 로그를 모두 보관할 수 있을 만큼 커야 합니다. 즉, Fast Recovery Area의 크기는 백업 하나와 여러 개의 아카이브된 로그를 보관할 수 있도록 최소한 데이터베이스 크기의 두 배여야 합니다.

Fast Recovery Area의 공간 관리는 백업 Retention 정책에 의해 제어됩니다. Retention 정책은 파일이 폐기되는 시기, 즉 데이터 Recovery 목표를 충족하는 데 데이터가 더 이상 필요하지 않은 시기를 결정합니다. 오라클 데이터베이스는 더 이상 필요하지 않은 파일을 삭제하여 이 저장 영역을 자동으로 관리합니다.

콘트롤 파일 다중화

데이터베이스 Failure로부터 보호하려면 데이터베이스의 콘트롤 파일 복사본이 여러 개 있어야 합니다.

	ASM 저장 영역	파일 시스템 저장 영역
최적의 사용법	각 디스크 그룹당 하나의 복사본 (예: +DATA 및 +FRA)	각 디스크에서 적어도 두 개 이상의 복사본 (개별 디스크 컨트롤러당 적어도 하나 이상)
추가 콘트롤 파일 생성을 위한 단계	추가 콘트롤 파일 복사본이 필요하지 않습니다.	<ol style="list-style-type: none">1. <code>ALTER SYSTEM SET control_files 명령으로 SPFILE 을 변경합니다.</code>2. 데이터베이스를 종료합니다.3. 콘트롤 파일을 새 위치에 복사합니다.4. 데이터베이스를 열고 새 콘트롤 파일이 추가되었는지 확인합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

콘트롤 파일 다중화

콘트롤 파일은 데이터베이스의 구조를 설명하는 작은 Binary File입니다. 콘트롤 파일은 데이터베이스를 마운트하거나 열 때마다 Oracle 서버가 기록에 사용할 수 있어야 합니다. 이 파일이 없으면 데이터베이스가 마운트될 수 없으므로 콘트롤 파일을 Recovery하거나 재생성해야 합니다. 데이터베이스에서 하나의 콘트롤 파일이 손상된 경우 그 영향을 최소화할 수 있도록 두 개 이상의 콘트롤 파일을 다른 저장 영역 장치에 두는 것이 좋습니다. 단일 콘트롤 파일이 손실되면 모든 콘트롤 파일을 항상 사용할 수 있어야 하므로 Instance가 실패합니다. 그러나 Recovery 시에는 다른 콘트롤 파일 중 하나를 복사하기만 하면 됩니다. 모든 콘트롤 파일이 손실되면 Recovery하기가 약간 더 어려워지지만 불가능에 가까울 정도로 작업이 어려운 것은 아닙니다.

콘트롤 파일 다중화(계속)

콘트롤 파일 추가

저장 영역 기술로 ASM을 사용할 경우 콘트롤 파일 두 개가 각 디스크 그룹(예: +DATA 및 +FRA)에 하나씩 있는 한 추가 다중화 작업이 필요하지 않습니다. OMF를 사용하는 데이터베이스(예: ASM 저장 영역을 사용하는 데이터베이스)에서는 RMAN(또는 Enterprise Manager)을 사용하여 Recovery 프로세스 중에 모든 추가 콘트롤 파일을 생성해야 합니다. 일반 파일 시스템 저장 영역을 사용하는 데이터베이스에서는 콘트롤 파일을 수동으로 추가해야 합니다.

1. 다음 명령을 사용하여 SPFILE을 변경합니다.

```
ALTER SYSTEM SET control_files =
  '/u01/app/oracle/oradata/orcl/control01.ctl' ,
  '/u02/app/oracle/oradata/orcl/control02.ctl' ,
  '/u03/app/oracle/oradata/orcl/control03.ctl' SCOPE=SPFILE;
```

2. 데이터베이스를 종료합니다.
3. 운영 체제를 사용하여 새 파일에 대해 선택한 위치로 기존 콘트롤 파일을 복사합니다.
4. 데이터베이스를 엽니다.

주: RMAN 사용에 대한 자세한 내용은 *Oracle Database 11g: Administration Workshop II* 과정 및 온라인 제품 설명서에서 다룹니다.

리두 로그 파일

리두 로그 그룹을 다중화하여 Media failure와 데이터 손실로부터 보호합니다. 그러면 데이터베이스 I/O가 높아집니다. 리두 로그 그룹에 대해서는 다음을 보관하는 것이 좋습니다.

- 그룹마다 최소한 두 개의 멤버(파일) 구성
- 각 멤버 구성 위치:
 - 파일 시스템 저장 영역을 사용할 경우 개별 디스크 또는 컨트롤러에서 구성
 - ASM을 사용하는 경우 개별 디스크 그룹 (예: +DATA 및 +FRA)에서 구성



주: 리두 로그를 다중화하면 전체 데이터베이스 성능에 영향을 줄 수 있습니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

리두 로그 파일

리두 로그 그룹은 하나 이상의 리두 로그 파일로 구성됩니다. 한 그룹의 각 로그 파일은 다른 로그 파일을 복제(Duplicate)한 것입니다. Oracle Corporation은 리두 로그 그룹에 그룹당 적어도 두 개의 파일을 구성할 것을 권장합니다. 파일 시스템 저장 영역을 사용할 경우 단일 장비의 Failure로 인해 전체 로그 그룹이 손상되지 않도록 개별 디스크 또는 컨트롤러에 각 멤버를 분배해야 합니다. ASM 저장 영역을 사용할 경우에는 개별 디스크 그룹(예: +DATA 및 +FRA)에 각 멤버를 두어야 합니다.

현재의 전체 로그 그룹이 손실되면 데이터 손실을 초래할 수 있으므로 가장 심각한 Media Failure 중 하나입니다. 다중 멤버로 구성된 로그 그룹에서의 단일 멤버의 손실은 사소한 문제이므로 데이터베이스 작업에 영향을 주지 않고 Alert log에 Alert만 게시(Publish)됩니다. 손실된 전체 로그 그룹의 Recovery는 고급 Recovery 기술을 필요로 하며, 이에 대해서는

Oracle Database 11g: Administration Workshop II 과정에서 설명합니다.

트랜잭션 정보가 로그에 기록되기 전에는 커밋을 완료할 수 없으므로 리두 로그 다중화는 데이터베이스 성능에 상당한 영향을 미칩니다. 따라서 리두 로그 파일은 가장 빠른 컨트롤러에서 제공하는 가장 빠른 디스크에 두어야 합니다. 자동 저장 영역 관리(ASM)를 사용하는 경우가 아니면, 가능한 한 다른 데이터베이스 파일을 리두 로그 파일과 동일한 디스크에 배치하지 마십시오. 주어진 시간에 하나의 그룹만 기록되므로 동일한 디스크에 여러 그룹의 멤버를 두는 것은 성능에 영향을 주지 않습니다.

리두 로그 다중화

Storage Type File System인 경우 File Name 및 File Directory 를 입력하라는 메시지가 표시됩니다.

Redo Log Groups

Edit Redo Log Group: 1: Add Redo Log Member

Storage Type	Automatic Storage Management
* DiskGroup	DATA
Template	<Default>
Alias Directory	
Alias Name	
<input type="checkbox"/> Reuse File	

Add

Select File Name

File Directory
+DATA/orcl/onlinelog/
group_1.261.689304441
+FRA/orcl/onlinelog/
group_1.257.689304447

Redo Log Members

Select	Group	Status	# of Members	Archived	Size (KB)	Sequence	First Change#
<input checked="" type="radio"/>	1	Active	2	No	51200	7	834285
<input type="radio"/>	2	Active	2	No	51200	8	849739
<input type="radio"/>	3	Current	2	No	51200	9	849745

Edit

Copyright © 2010, Oracle. All rights reserved.

리두 로그 다중화

기존 로그 그룹에 멤버를 추가하여 리두 로그를 다중화할 수 있습니다. 데이터베이스가 열린 상태에서 유저 성능에 영향을 주지 않고 리두 로그 그룹에 멤버를 추가하려면 다음 단계를 수행하십시오.

- Enterprise Manager > Server > Redo Log Groups를 선택합니다.
- 그룹을 선택하고 Edit 버튼을 누르거나, 그룹 번호 링크를 누릅니다.
Edit Redo Log Group 페이지가 나타납니다.
- Redo Log Members 영역에서 Add를 누릅니다.
Add Redo Log Member 페이지가 나타납니다.
- 적합한 Storage Type을 선택하고 필요한 정보를 입력합니다. ASM의 경우 디스크 그룹을 선택하고, 필요에 따라 템플릿 및 Alias 정보를 지정합니다. File System 저장 영역의 경우 파일 이름과 파일 디렉토리를 입력합니다. Continue를 누릅니다.

다중화하려는 모든 기존 그룹에 대해 이 단계를 반복합니다. 다음은 리두 로그 그룹 1에 리두 로그 멤버를 추가하는(ASM 사용) SQL 구문을 보여 주는 예입니다.

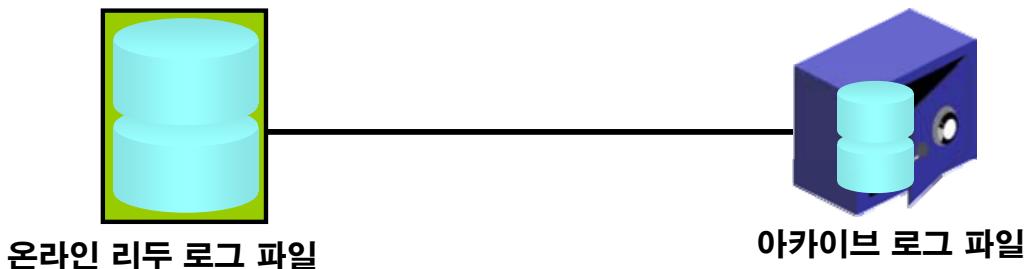
```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA' TO GROUP 1;
```

리두 로그 멤버를 그룹에 추가하면 멤버의 상태가 INVALID로 표시되며 이는 V\$LOGFILE 뷰에서 확인할 수 있습니다. 이는 그룹의 새 멤버가 아직 기록되지 않았기 때문입니다. 로그 스위치가 있고 새 멤버를 포함하는 그룹이 CURRENT가 되면 멤버의 상태가 NEL로 변경됩니다.

아카이브 로그 파일

리두 정보를 보존하려면 다음 단계를 수행하여 리두 로그 파일의 아카이브된 복사본을 생성합니다.

1. 아카이브 로그 파일의 이름 지정 규칙을 지정합니다.
2. 하나 이상의 아카이브 로그 파일 위치를 지정합니다.
3. 데이터베이스를 ARCHIVELOG 모드로 전환합니다.



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

아카이브 로그 파일

Instance는 온라인 리두 로그 그룹을 트랜잭션 정보를 저장할 순환 버퍼로 취급하여 한 그룹을 채운 후 다음 그룹으로 이동합니다. 모든 그룹이 기록되면 Instance는 첫번째 로그 그룹에 있는 정보를 겹쳐쓰기 시작합니다.

recovery 가능성을 최대화하도록 데이터베이스를 구성하려면 온라인 리두 로그 그룹을 겹쳐쓰기 전에 해당 복사본을 만들도록 데이터베이스를 지정해야 합니다. 이러한 복사본이 아카이브된 로그입니다.

아카이브 로그 파일을 간편하게 생성하려면 다음과 같이 하십시오.

1. 아카이브 로그에 대해 이름 지정 규칙을 지정합니다.
2. 아카이브 로그를 저장할 대상을 지정합니다. 대상 중 하나는 Fast Recovery Area일 수 있습니다.
3. 데이터베이스를 ARCHIVELOG 모드로 설정합니다.

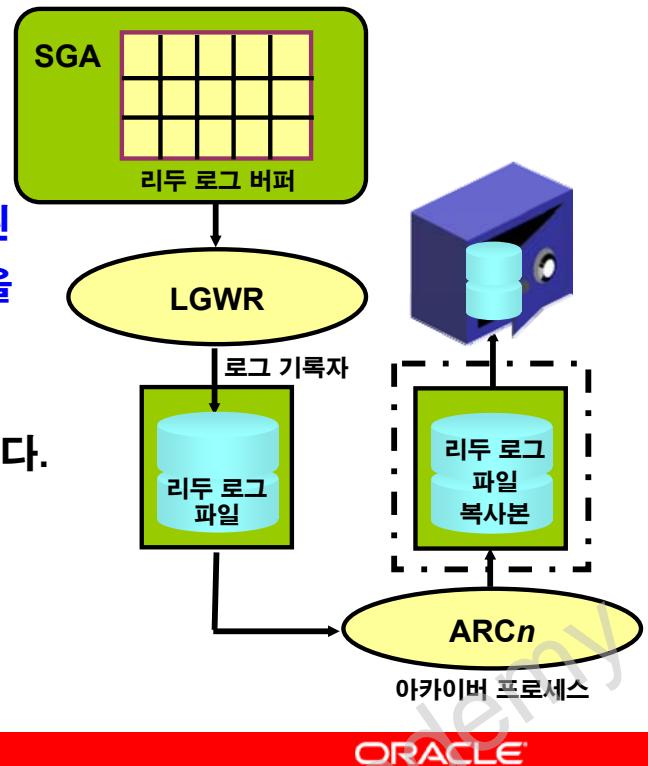
주: Fast Recovery Area를 사용하는 경우에는 단계 1 및 2를 수행할 필요가 없습니다.

데이터베이스를 ARCHIVELOG 모드로 설정하려면 대상이 존재해야 합니다. 대상으로 디렉토리가 지정된 경우 디렉토리 이름 끝에 슬래시(/)가 있어야 합니다.

ARCn(아카이버 프로세스)

ARCn(아카이버 프로세스):

- 선택적인 백그라운드 프로세스입니다.
- 데이터베이스에 대해 ARCHIVELOG 모드가 설정된 경우 온라인 리두 로그 파일을 자동으로 아카이브합니다.
- 데이터베이스에 수행한 모든 변경 사항 레코드를 보존합니다.



Copyright © 2010, Oracle. All rights reserved.

ARCn(아카이버 프로세스)

ARCn은 선택적인 백그라운드 프로세스입니다. 그러나 이 프로세스는 디스크 손실 후의 데이터베이스 Recovery에 매우 중요합니다. 온라인 리두 로그 그룹이 가득 차면 Oracle Instance는 다음 온라인 리두 로그 그룹에 기록하기 시작합니다. 한 온라인 리두 로그 그룹에서 다른 온라인 리두 로그 그룹으로 전환하는 프로세스를 로그 스위치라고 합니다. ARCn 프로세스는 로그 스위치가 발생할 때마다 가득 찬 로그 그룹을 아카이브하기 시작합니다. 이 프로세스는 온라인 리두 로그 그룹을 재사용할 수 있도록 자동으로 아카이브하므로 데이터베이스에 수행한 모든 변경 사항이 보존됩니다. 따라서 디스크 드라이브가 손상된 경우에도 failure 시점으로 데이터베이스 recovery가 가능합니다.

데이터베이스를 ARCHIVELOG 모드와 NOARCHIVELOG 모드 중에서 어느 모드에서 작동하도록 구성할지 여부는 DBA가 결정해야 하는 중요한 문제 중 하나입니다.

- NOARCHIVELOG 모드에서는 로그 스위치가 발생할 때마다 온라인 리두 로그 파일을 겹쳐씁니다.
- ARCHIVELOG 모드에서는 먼저 가득 찬 온라인 리두 로그 파일의 비활성 그룹을 아카이브해야 해당 로그 파일을 다시 사용할 수 있습니다.

주:

- ARCHIVELOG 모드는 대부분의 백업 전략에 반드시 필요하며 구성하기 매우 쉽습니다.
- 아카이브 로그 파일 대상이 모두 채워졌거나 쓸 수 없는 경우 데이터베이스가 결국 중단됩니다. 아카이브 로그 파일 대상에서 아카이브 파일을 제거하면 데이터베이스가 다시 작동됩니다.

아카이브 로그 파일: 이름 지정 및 대상

Recovery Settings 페이지에서 이름 지정 및 아카이브 대상 정보를 지정합니다. 파일 시스템 저장 영역을 사용하는 경우 각 디스크에 걸쳐 여러 위치를 추가하는 것이 좋습니다.

Media Recovery

The database is currently in ARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time are possible, but you must provide space for archived redo log files. If you change the database to ARCHIVELOG mode, you should perform a backup immediately. In NOARCHIVELOG mode, only cold backups are possible and data may be lost in the event of database corruption.

ARCHIVELOG Mode*

Log Archive Filename Format*

Number	Archived Redo Log Destination	Status	Type
1	USE_DB_RECOVERY_FILE_DEST	VALID	Local

TIP It is recommended that archived redo log files be written to multiple locations spread across the different disks.
 TIP You can specify up to 10 archived redo log destinations.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

아카이브 로그 파일: 이름 지정 및 대상

아카이브 로그 파일 이름 및 대상을 구성하려면 Enterprise Manager > Availability > Configure Recovery Settings를 선택합니다.

이전 로그 파일을 겹쳐쓰지 않도록 하려면 각각의 아카이브 로그 파일이 고유한 이름을 가져야 합니다. 슬라이드에 표시된 것과 같이 이름 지정 형식을 지정하십시오. 고유한 파일 이름을 생성할 수 있도록 Oracle Database 11g에서는 이름 형식에 다양한 대체 문자를 허용합니다.

- **%s:** 파일 이름의 일부로 Log Sequence Number를 포함합니다.
- **%t:** 파일 이름의 일부로 스레드 번호를 포함합니다.
- **%r:** resetlogs ID를 포함하여 Log Sequence Number를 재설정하는 특정 고급 Recovery 기법후에도 아카이브 로그 파일 이름이 고유하게 유지되도록 보장합니다.
- **%d:** 파일 이름의 일부로 데이터베이스 ID를 포함합니다.

형식에는 최적의 사용법으로 %s, %t 및 %r이 포함됩니다(%d는 여러 데이터베이스가 동일한 아카이브 로그 대상을 공유하는 경우에 포함될 수도 있음).

기본적으로 Fast Recovery Area가 활성화된 경우 USE_DB_RECOVERY_FILE_DEST는 아카이브 로그 파일 대상으로 지정됩니다. 아카이브 로그 파일은 최대 10개의 다른 대상에 기록될 수 있습니다. 대상은 로컬(디렉토리) 또는 원격(Standby Database에 대한 Oracle Net alias)에 있을 수 있습니다. 더 많은 대상을 추가하려면 Add Another Row를 누릅니다. Recovery 설정을 변경하려면 SYSDBA 또는 SYSOPER로 연결해야 합니다.

주: 아카이브를 이 위치로 보내지 않으려면 USE_DB_RECOVERY_FILE_DEST를 삭제하십시오.

ARCHIVELOG 모드 활성화

데이터베이스를 ARCHIVELOG 모드로 설정하려면 Enterprise Manager에서 다음 단계를 수행하십시오.

1. ARCHIVELOG Mode 체크 박스를 선택하고 Apply를 누릅니다.
데이터베이스는 MOUNT 상태에서만 ARCHIVELOG 모드로 설정될 수 있습니다.
2. SYSDBA 권한을 사용하여 데이터베이스를 재시작합니다.
3. (선택 사항) 아카이브 상태를 확인합니다.
4. 데이터베이스를 백업합니다.

주: ARCHIVELOG 모드의

데이터베이스는 모든 범위의 백업 및 Recovery 옵션에 액세스할 수 있습니다.

```
sqlplus / as sysdba
shutdown immediate
startup mount
alter database archivelog;
alter database open;
archive log list
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

ARCHIVELOG 모드 활성화

1. Enterprise Manager에서 Availability > Configure Recovery Settings > ARCHIVELOG Mode를 선택합니다. 상응하는 SQL 명령은 다음과 같습니다.
SQL> ALTER DATABASE ARCHIVELOG;
이 명령은 데이터베이스가 MOUNT 상태일 때만 실행할 수 있습니다. 그러므로 이 마지막 단계를 완료하려면 Instance를 재시작해야 합니다.
2. Enterprise Manager에는 데이터베이스를 재시작하는 동안 운영 체제와 데이터베이스 인증서를 요청하는 메시지가 표시됩니다. 데이터베이스 인증서는 반드시 SYSDBA 권한을 가진 유저의 인증서여야 합니다.
3. Instance가 재시작되면 아카이브 프로세스, 로그 형식 및 로그 대상에 대한 변경 사항이 적용됩니다. SQL*Plus에서는 ARCHIVE LOG LIST 명령을 사용하여 해당 변경 사항을 확인할 수 있습니다.
4. ARCHIVELOG 모드에서 수행된 마지막 백업에서만 데이터베이스를 Recovery 할 수 있기 때문에 해당 모드로 전환한 후 데이터베이스를 백업합니다.

데이터베이스를 NOARCHIVELOG 모드(기본값)로 두면 마지막 백업 시점까지만 recovery가 가능합니다. 해당 백업 이후에 수행된 모든 트랜잭션은 손실됩니다.

ARCHIVELOG 모드에서는 마지막 커밋 시점까지 Recovery가 가능합니다. 대부분의 운영 중인 데이터베이스는 ARCHIVELOG 모드로 실행됩니다.

퀴즈

Statement failure는 의도적으로는 절대로 발생할 수 없으며 항상 DBA가 문제를 해결해야 합니다.

- 1. 참**
- 2. 거짓**



Copyright © 2010, Oracle. All rights reserved.

해답: 2

퀴즈

Fast Recovery Area를 구성하는 파라미터는 무엇입니까?

1. **FLASH_RECOVERY_AREA_SIZE**
2. **DB_RECOVERY_FILE_DEST**
3. **FLASH_RECOVERY_AREA_LOC**
4. **DB_RECOVERY_FILE_DEST_SIZE**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 2, 4

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 오라클 데이터베이스에서 발생할 수 있는 Failure 유형 식별
- Instance recovery 튜닝 방법 설명
- 체크포인트, 리두 로그 파일 및 아카이브 로그 파일의 중요성 파악
- Fast Recovery Area 구성
- ARCHIVELOG 모드 구성

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

연습 14 개요: Recovery 가능성을 위한 구성

이 연습에서는 다음 내용을 다룹니다.

- 콘트롤 파일 확인
- 기본 Fast Recovery Area 구성
- 리두 로그 그룹 다중화
- 데이터베이스를 ARCHIVELOG 모드로 설정
- 중복 아카이브 로그 생성 확인

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 백업 수행

15

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- 일관성 있는 데이터베이스 백업 생성
- 데이터베이스를 종료하지 않고 백업
- Incremental 백업 생성
- 데이터베이스 백업 자동화
- 백업 관리 및 백업 보고서 보기
- Fast Recovery Area 모니터

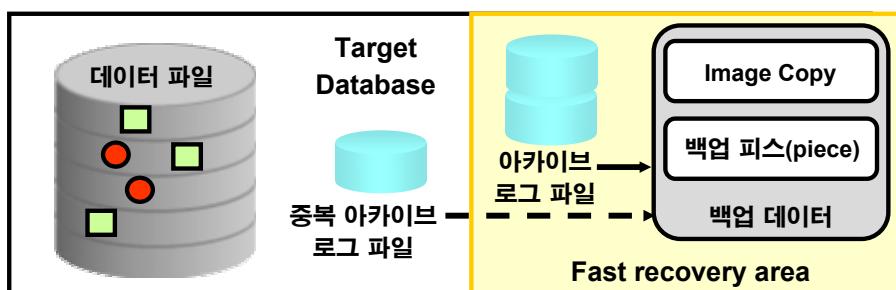
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

백업 솔루션: 개요

다음을 사용하여 백업을 수행할 수 있습니다.

- Recovery Manager
- Oracle Secure Backup
- User-managed 방식의 백업



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

백업 솔루션: 개요

이 단원의 뒷부분에서 자세히 설명하겠지만 오라클 데이터베이스의 백업 방식으로는 RMAN(Recovery Manager)이 권장됩니다.

Oracle Secure Backup은 테이프 백업 및 운영 체제 파일의 백업 기능을 추가하여 기존 기능을 보완합니다.

User-managed 방식의 백업은 DBA가 작성해야 하는 스크립트를 기반으로 합니다. 이 옵션은 많은 수작업이 필요하기 때문에 점차 사용이 줄어들고 있습니다.

Oracle Secure Backup

- Oracle Secure Backup 및 RMAN은 Oracle 환경에 다음과 같은 엔드 투 엔드 백업 솔루션을 제공합니다.
 - 파일 시스템 데이터 및 오라클 데이터베이스에 대한 중앙화된 테이프 백업 관리
 - 가장 잘 통합된 RMAN 백업용 Media Management 계층
 - 네트워크상의 데이터 백업
- 전체 백업 솔루션에 대한 단일 기술 지원 리소스로 문제 해결이 가능합니다.
- 간단하면서 비용이 저렴한 신뢰할 수 있는 데이터 보호를 보장합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Secure Backup

Oracle의 최신 데이터베이스 백업 및 recovery 제품은 Recovery Manager입니다. Oracle Secure Backup은 다음과 같은 방식으로 기존의 기능을 보완합니다.

- **전체 백업 솔루션:** Oracle Secure Backup은 데이터베이스 데이터와 비데이터베이스 데이터를 모두 보호함으로써 전체 Oracle 환경을 보호합니다.
- **Media Management:** Oracle Secure Backup은 테이프로의 RMAN 데이터베이스 백업을 위한 Media Management 계층을 제공합니다. Oracle Secure Backup 이전에는 RMAN 테이프 백업과의 통합을 제공하는 비싼 Third-party Media Management 제품을 구매해야 했습니다.
- **네트워크상의 데이터 백업 지원:** Oracle Secure Backup은 네트워크 연결 컴퓨터 시스템의 데이터를 네트워크상의 다른 저장 리소스에 백업합니다. Oracle Secure Backup은 서버, 클라이언트, NAS(Network Attached Storage) 서버 및 기타 저장 장치 등이 포함되는 다양한 구성을 지원하며, 네트워크 저장 영역 환경을 보호합니다.

RMAN과 Oracle Secure Backup을 결합하면 Oracle 제품 스택 전체에 적용되는 엔드 투 엔드 백업 솔루션이 됩니다. Oracle은 전체 백업 솔루션을 취급하므로 이 솔루션을 사용하면 보다 향상된 고객 지원이 이뤄질 수 있습니다.

User-managed 방식의 백업

user-managed 방식의 시나리오는 다음과 같습니다.

- 백업 필요성과 상태를 추적하는 수동 프로세스.
- 일반적으로 직접 작성한 스크립트 사용
- 데이터베이스 파일을 올바른 백업 모드로 설정
- 운영 체제 명령을 사용하여 파일 백업

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

User-managed 방식의 백업

User-managed 방식의 백업은 대화식으로 수행할 수 있습니다. 그러나 대부분의 경우 백업을 수행하기 위해서는 스크립트를 직접 작성해야 합니다. 여기에는 몇 가지 시나리오가 있는데 이러한 시나리오를 처리할 스크립트를 작성해야 합니다.

스크립트를 통해 수행해야 하는 작업은 다음과 같습니다.

- V\$DATAFILE을 query하여 백업이 필요한 데이터 파일과 해당 파일의 현재 상태 파악
- V\$LOGFILE을 query하여 온라인 리두 로그 파일 식별
- V\$CONTROLFILE을 query하여 백업할 콘트롤 파일 식별
- 각 테이블스페이스를 온라인 백업 모드로 둠
- V\$BACKUP을 query하여 어떤 데이터 파일이 온라인 백업 모드에 놓인 테이블스페이스에 속하는지 확인
- 운영 체제 복사 명령을 실행하여 데이터 파일을 백업 위치에 복사
- 각 테이블스페이스의 온라인 백업 모드 해제

용어

- 백업 전략에는 다음이 포함됩니다.
 - 전체 데이터베이스(Whole)
 - 데이터베이스의 일부(Partial)
- 백업 유형이 포함하는 정보는 다음을 나타냅니다.
 - 선택한 파일 내의 모든 데이터 블록(Full)
 - 이전 백업 이후 변경된 정보만(Incremental)
 - Cumulative(마지막 레벨 0부터의 변경 사항)
 - Differential(마지막 incremental부터의 변경 사항)
- 백업 모드는 다음과 같습니다.
 - 오프라인(일관성 있음, 콜드)
 - 온라인(일관성 없음, 핫)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

용어

Whole Database Backup: 모든 데이터 파일과 하나 이상의 콘트롤 파일(단일 데이터베이스 내의 모든 콘트롤 파일은 동일함)이 포함됩니다.

Partial 데이터베이스 백업: 0개 이상의 테이블스페이스 및 0개 이상의 데이터 파일이 포함될 수 있으며 콘트롤 파일은 포함될 수도 있고 포함되지 않을 수도 있습니다.

Full 백업: 백업되는 파일 내에서 데이터를 포함하는 각 데이터 블록의 복사본을 만듭니다.

Incremental 백업: 이전 백업 이후 변경된 모든 데이터 블록의 복사본을 만듭니다. 오라클 데이터베이스에서는 두 가지 레벨의 Incremental 백업(0 및 1)을 지원합니다. 레벨 1 Incremental 백업의 유형은 *cumulative* 또는 *differential* 중 하나일 수 있습니다. Cumulative 백업은 마지막 레벨 0 백업 이후에 발생한 모든 변경 사항을 백업합니다. Differential 백업은 마지막 Incremental 백업(레벨 0 백업 또는 레벨 1 백업일 수 있음) 이후에 발생한 모든 변경 사항을 백업합니다. RMAN을 사용한 변경 사항 추적은 incremental 백업을 지원합니다.

오프라인 백업("콜드" 또는 일관성 있는 백업이라고도 함): 데이터베이스가 닫혀 있는 동안 수행됩니다. 백업 시점에 데이터 파일 헤더의 시스템 변경 번호(SCN)가 콘트롤 파일의 SCN과 일치하므로 이 백업은 일관성이 있습니다.

온라인 백업("핫" 또는 일관성 없는 백업이라고도 함): 데이터베이스가 열려 있는 동안 수행됩니다. 데이터베이스가 열린 상태에서는 데이터 파일이 콘트롤 파일과 동기화되었는지 보장할 수 없으므로 이 백업은 일관성이 없습니다.

용어

백업은 다음 형태로 저장됩니다.

- **Image Copy**
- **백업 셋**

데이터 파일 #1
데이터 파일 #2
데이터 파일 #3
데이터 파일 #4
데이터 파일 #5
데이터 파일 #6

Image Copy
(OS 형식의 복제 데이터 및 로그 파일)

데이터 파일 #1	데이터 파일 #2
데이터 파일 #3	데이터 파일 #4
데이터 파일 #5	데이터 파일 #6

백업 셋
(오라클 고유 형식의 압축된 Binary File)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

용어(계속)

Image Copy: 데이터 또는 아카이브된 로그 파일을 복제(Duplicate)한 것으로, 운영 체제 명령을 사용하여 단순히 파일을 복사하는 것과 유사합니다.

백업 셋: 여러 데이터 파일, 콘트롤 파일, 서버 파라미터 파일 또는 아카이브된 로그 파일이 포함된 하나 이상의 Binary File 모음입니다. 백업 셋을 사용하면 빈 데이터 블록이 저장되지 않으므로 백업 셋에서 소모하는 디스크나 테이프의 공간이 줄어듭니다. 백업 셋을 압축하면 백업에 필요한 공간을 더 줄일 수 있습니다.

Image Copy는 디스크에 백업해야 합니다. 백업 셋은 디스크에 보내거나 테이프에 직접 보낼 수 있습니다.

Image Copy로 백업을 생성할 때의 이점은 보다 세분화된 복원 작업을 수행할 수 있다는 것입니다. Image copy를 사용할 경우 백업 위치에서 필요한 파일만 검색하면 됩니다.

백업 셋을 사용하는 경우 필요한 파일을 추출하려면 먼저 전체 백업 셋을 백업 위치에서 검색해야 합니다.

백업 셋으로 백업을 생성할 때의 이점은 공간 사용률이 향상된다는 것입니다. 대부분의 데이터베이스에서는 데이터 블록의 20% 이상이 빈 블록입니다. Image Copy는 데이터 블록이 비어 있더라도 모든 데이터 블록을 백업합니다. 백업 셋은 백업에 필요한 공간을 대폭 줄입니다. 대부분의 시스템에서는 Image Copy보다 백업 셋이 유용합니다.

RMAN(Recovery Manager)

- 강력한 제어 및 스크립팅 언어
- Enterprise Manager와 통합
- 대부분의 범용 백업 소프트웨어와 상호 작용하는 Published API
- 데이터, 콘트롤, 아카이브된 로그 및 서버 파라미터 파일 백업
- 디스크나 테이프에 파일 백업



ORACLE

Copyright © 2010, Oracle. All rights reserved.

RMAN(Recovery Manager)

RMAN은 백업 및 Recovery 작업에 사용되는 오라클 데이터베이스의 구성 요소입니다.

RMAN은 일관성 있는 백업과 일관성 없는 백업을 수행하고, Incremental 또는 Full 백업을 수행하며, Whole 데이터베이스 또는 데이터베이스의 일부를 백업할 수 있습니다.

RMAN은 자체의 강력한 작업 제어 및 스크립팅 언어를 사용하여 대부분의 범용 백업 소프트웨어 솔루션과 RMAN을 연결하는 Published API를 사용합니다.

RMAN은 신속한 recovery를 위해 디스크에 백업을 저장하거나 장기간 저장을 위해 테이프에 백업을 저장합니다. RMAN을 통해 테이프에 백업을 저장하려면 Oracle Secure Backup을 사용하거나 MML(Media Management Library)이라는 테이프 장치에 대한 인터페이스를 구성해야 합니다.

Enterprise Manager는 가장 일반적으로 사용되는 RMAN 기능을 그래픽 인터페이스로 제공합니다. 고급 백업 및 recovery 작업은 RMAN의 명령행 클라이언트를 통해 액세스할 수 있습니다.

RMAN의 고급 기능에 대한 자세한 사항은 *Oracle Database 11g: Administration Workshop II* 과정이나 *Oracle Backup and Recovery User's Guide*를 참조하십시오.

백업 설정 구성

The screenshot shows the 'Backup Settings' page in Oracle Enterprise Manager. The 'Device' tab is selected under 'Disk Settings'. It includes fields for 'Parallelism' (set to 1), 'Disk Backup Location' (Fast Recovery Area), and 'Disk Backup Type' (Backup Set). The 'Backup Set' tab is selected, showing 'Maximum Backup Piece (File)' size (1 MB) and compression algorithm (BZIP2). The 'Policy' tab is also visible. Below these are sections for 'Tape Settings' (Copies of Datafile Backups, Copies of Archivelog Backups) and 'Host Credentials' (Username and Password).

ORACLE

Copyright © 2010, Oracle. All rights reserved.

백업 설정 구성

Enterprise Manager > Availability > Backup Settings를 선택합니다. 여기서는 백업을 생성하는데 사용되는 지속 백업 설정을 관리할 수 있습니다. 설정은 디스크와 테이프에 대해 별도로 구성됩니다. 테이프 설정은 MML(Media Management Library) 기능에 의존합니다. 디스크 설정에는 다음 내용이 포함됩니다.

- Parallelism:** 생성할 백업 정보에 대한 별도의 스트림 수를 지정합니다. 최적의 병렬화 설정은 하드웨어에 따라 다릅니다. 하드웨어 리소스가 늘어나면 병렬도도 적절히 높아집니다. 일반적으로는 디스크 백업 위치가 스트라이핑되는 디스크 수로 병렬화를 설정해야 합니다. 테이프 백업의 경우에는 보유한 테이프 드라이브 수와 동일하게 병렬화를 설정해야 합니다.
- Disk Backup Location:** 백업을 저장할 위치를 지정합니다. 기본 위치는 Fast Recovery Area입니다. 이를 변경하려면 Test Disk Backup을 눌러 RMAN이 새 위치에 쓸 수 있는지 확인합니다.
- Disk Backup Type:** Backup Set, Compressed Backup Set 또는 Image Copy를 선택합니다.

Backup Set 탭을 눌러 백업 피스(piece)의 최대 파일 크기를 설정하고, 압축된 백업 셋에 사용할 압축 알고리즘을 지정하고, 테이프 백업에 대한 중복성을 지정합니다. Enterprise Manager에서 백업 설정에 대한 변경 사항을 저장하려면 호스트 인증서(Host credentials)가 필요합니다.

백업 설정 구성

Backup Settings

Device Backup Set Policy

Backup Policy

Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change

Autobackup Disk Location
An existing directory or diskgroup name where the control file and server parameter file will be backed up. If you do not specify a location, the files will be backed up to the flash recovery area location.

Optimize the whole database backup by skipping unchanged files such as read-only and offline datafiles that have been backed up

Enable block change tracking for faster incremental backups

Block Change Tracking File
Specify a location and file, otherwise an Oracle managed file will be created in the database area.

Tablespaces Excluded From Whole Database Backup
Populate this table with the tablespaces you want to exclude from a whole database backup. Use the Add button to add tablespace.

Select Tablespace Name	Tablespace Number	Status	Contents
No Items Selected			

TIP These tablespaces can be backed up separately using tablespace backup.

Retention Policy

Retain All Backups
You must manually delete any backups

Retain backups that are necessary for a recovery to any time within the specified number of days (point-in-time recovery)
Days: 31
Recovery Window

Retain at least the specified number of full backups for each datafile
Backups: 1
Redundancy

Archived Redo Log Deletion Policy
Specify the deletion policy for archived redo log files. The archived redo log files will be eligible for deletion if the flash recovery area becomes full.

None
If a flash recovery area is set, archived redo log files that have been backed up to a tertiary device and are obsolete based on the retention policy will be deleted.

Delete archived redo log files after they have been backed up the specified number of times
Backups: 1

Host Credentials
To save the backup settings, supply operating system login credentials to access the target database.

* Username: _____
* Password: _____
 Save as Preferred Credential

최적의 사용법



ORACLE

Copyright © 2010, Oracle. All rights reserved.

백업 설정 구성(계속)

Policy 탭을 눌러 다음을 수행합니다.

- 각 백업마다 콘트롤 파일과 서버 파라미터 파일(SPFILE)을 자동으로 백업합니다. 또한 이러한 백업을 Fast Recovery Area로 이동하지 않으려면 직접 백업 위치를 지정할 수 있습니다.
- 이미 보관된 백업의 파일 부분과 정확히 일치하는 파일은 백업하지 않음으로써 백업을 최적화합니다. 이렇게 설정하면 읽기 전용 파일과 오프라인 데이터 파일을 건너뛸 수 있습니다.
- 블록 변경 사항 추적(Block Change Tracking)을 활성화하고 추적 파일의 위치를 지정합니다. Incremental 백업을 생성하려는 경우 이러한 설정을 사용하면 Incremental 백업에 포함할 블록 선택 시 필요한 시간을 줄일 수 있습니다.
- Whole Database Backup에서 테이블스페이스를 제외합니다. 일부 관리자는 쉽게 재생성할 수 있는 데이터나 객체(자주 일괄 로드되는 인덱스나 데이터 등)를 포함하는 테이블스페이스는 백업하지 않습니다.
- retention 정책을 지정합니다. RMAN에서 백업을 얼마나 오래 보관할지 지정합니다. 백업 저장 위치로 Fast Recovery Area를 사용하는 경우 RMAN은 Retention 정책의 허용 범위 내에서 새 백업에 필요한 공간 확보를 위해 오래된 백업을 자동으로 삭제합니다. 기본적으로 마지막 백업만 보관됩니다. Retention 정책은 백업 수 또는 일 수로 지정할 수 있습니다.

백업 일정 잡기: 전략

Schedule Backup

Oracle provides an automated backup strategy based on your disk and/or tape configuration. Alternatively, you can implement your own customized backup strategy.

Oracle-Suggested Backup

Schedule a backup using Oracle's automated backup strategy. [Schedule Oracle-Suggested Backup](#)

This option will back up the entire database. The database will be backed up on daily and weekly intervals.

Customized Backup

Select the object(s) you want to back up. [Schedule Customized Backup](#)

Whole Database
 Tablespaces
 Datafiles
 Archived Logs
 All Recovery Files on Disk
Includes all archived logs and disk backups that are not already backed up to tape.

Host Credentials

To perform a backup, supply operating system login credentials to access the target database.

* Username: oracle
* Password:
 Save as Preferred Credential

Backup Strategies

Oracle-suggested:

- Provides an out-of-the-box backup strategy based on the backup destination
- Sets up recovery window for backup management
- Schedules recurring and immediate backups
- Automates backup management

Customized:

- Specify the objects to be backed up
- Choose disk or tape backup destination
- Override the default backup settings
- Schedule the backup

ORACLE

Copyright © 2010, Oracle. All rights reserved.

백업 일정 잡기: 전략

Enterprise Manager > Availability > Schedule Backup을 선택합니다. Oracle 권장 백업 전략이나 유저 고유의 커스터마이즈된 전략을 선택합니다. Oracle 권장 백업 전략은 온라인으로 수행되는 1회의 Whole Database Backup입니다. 이 백업은 Baseline incremental 레벨 0 백업입니다. 자동화된 백업 전략은 이어서 각 다음 날짜에 대해 Incremental 레벨 1 백업의 일정을 잡습니다. 3일 이후로 RMAN은 일일 incremental 백업을 시작하기 전에 해당 날짜 n-1의 시작점에서 수행된 레벨 1 백업을 레벨 0 백업에 적용합니다.

Schedule Customized Backup을 누르면 보다 광범위한 구성 옵션에 액세스할 수 있습니다. 백업 할 객체로 Whole 데이터베이스(기본값), 개별 테이블스페이스, 데이터 파일, 아카이브된 로그 또는 현재 디스크에 상주하는 Oracle 백업(테이프로 이동할 예정)을 선택합니다.

이들 두 전략에서는 모두 암호화된(encrypted) 백업을 설정할 수 있습니다.

백업 일정 잡기: 옵션

Options Settings Schedule Review

Schedule Customized Backup: Options

Database **orcl.oracle.com** Cancel Step 1 of 4 Next

Backup Strategy **Customized Backup**

Object Type **Whole Database**

Backup Type

Full Backup
 Use as the base of an incremental backup strategy

Incremental Backup
A level 1 cumulative incremental backup includes all blocks changed since the most recent level 0 backup.
 Refresh the latest datafile copy on disk to the current time using the incremental backup

Backup Mode

Online Backup
Can be performed when the database is open.

Offline Backup
If the database is open at the time of backup, it will be shut down and mounted before the backup, then re-opened after the backup.

Advanced

Also back up all archived logs on disk
 Delete all archived logs from disk after they are successfully backed up

Delete obsolete backups
Delete backups that are no longer required to satisfy the retention policy.

Use proxy copy supported by media management software to perform a backup
If proxy copy of the selected files is not supported, a conventional backup will be performed.

Maximum Files per Backup Set

Section Size KB

Backing up large files in parallel, using sections of the specified size. (This parameter overrides Maximum Backup Piece Size in Backup Settings.)

►Encryption

ORACLE

Copyright © 2010, Oracle. All rights reserved.

백업 일정 잡기: 옵션

Full 또는 Incremental 백업 유형을 선택합니다. Full 데이터베이스 백업을 수행하는 경우 "Use as the base of an incremental backup strategy"를 선택하여 Full 데이터베이스 백업을 Incremental 레벨 0 백업으로 만들 수 있습니다. Image Copy를 사용할 경우 "Refresh the latest datafile copy on disk to the current time using the incremental backup"을 선택하여 새 Image Copy를 생성하는 대신 기존 백업을 갱신할 수 있습니다.

이 작업을 수행하면서 유저가 데이터베이스를 계속 사용하도록 하려면 Online Backup을 선택합니다. 유저가 액세스할 필요가 없는 경우에는 마운트된 Instance로 수행되는 "Offline Backup"을 선택합니다.

앞에서 구성한 retention 정책 범위를 벗어나는 백업을 제거하려면 "Delete obsolete backups"를 선택합니다. Fast Recovery Area에 백업하는 경우 RMAN은 불필요한 백업을 자동으로 제거합니다. 고급 옵션 및 암호화(encryption)에 대한 자세한 사항은 *Oracle Database 11g: Administration Workshop II* 과정과 백업 및 Recovery 설명서를 참조하십시오.

백업 일정 잡기: 설정

Options **Settings** Schedule Review

Schedule Customized Backup: Settings

Database **orcl.oracle.com** [Cancel](#) [Back](#) Step 2 of 4 [Next](#)

Backup Strategy **Customized Backup**
Object Type **Whole Database**

Select the destination media for this backup. You can also override the default backup settings.

Disk
 Tape

Disk Backup Location **+FRA**

Media Management Vendor (MMV) Library Parameters **Not specified**

[View Default Settings](#) [Override Default Settings](#)
Changed settings will only apply to the current backup.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

백업 일정 잡기: 설정

백업을 디스크에 옮길지 테이프에 옮길지 선택합니다.

정기적으로 일정이 잡힌 백업 외에도 1회 실행 백업을 생성하려면 **Override Current Settings**를 누르고 백업 설정을 지정합니다.

백업 일정 잡기: 일정

Options Settings **Schedule** Review

Schedule Customized Backup: Schedule

Database **orcl.oracle.com** **Cancel** **Back** **Step 3 of 4** **Next**

Backup Strategy **Customized Backup**

Object Type **Whole Database**

Job

* Job Name **BACKUP_ORCL.ORACLE.COM_001**

Job Description **Whole Database Backup**

Schedule

Type One Time (Immediately) One Time (Later) Repeating

Frequency Type **By Minutes**

Repeat Every Minutes

Time Zone **(UTC-08:00) US Pacific Time (PST)**

Start Date **Jun 18, 2009**

Start Time **1:00 AM**

Repeat Until Indefinite
 Specified Date

Date **Jun 18, 2009**

(example: Jun 18, 2009)

Time **1:00 AM**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

백업 일정 잡기: 일정

1회 작업 또는 자동화된 반복 프로세스 중에서 백업의 일정을 잡을 방법을 선택합니다.

Recovery 가능성을 최대화하도록 데이터베이스를 구성하려면 오라클은 정기적으로 일정이 잡힌 백업을 권장합니다. 백업을 자동화하면 관리자의 작업 로드를 간소화할 수 있습니다.

Repeating을 선택하면 페이지에 추가 일정 잡기 세부 정보가 표시됩니다.

백업 일정 잡기: 검토

Schedule Customized Backup: Review

Database **orcl.oracle.com** **Cancel** **Edit RMAN Script** **Back** **Step 4 of 4** **Submit Job**

Backup Strategy **Customized Backup**

Object Type **Whole Database**

Settings

Destination **Disk**
Backup Type **Full Backup**
Backup Mode **Online Backup**
Flash Recovery Area **+FRA**

RMAN Script

The RMAN script below is generated based on previous input.

```
backup device type disk tag '%TAG' database;
backup device type disk tag '%TAG' archivelog all not backed up;
```

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

백업 일정 잡기: 검토

RMAN은 고유한 명령 구문 및 스크립팅 언어를 사용합니다.

이 페이지를 사용하여 필요한 경우 RMAN 스크립트를 커스터마이즈하거나 기록을 위해 스크립트를 복사할 수 있습니다.

콘트롤 파일을 Trace File로 백업

콘트롤 파일에는 추가 백업 옵션이 있습니다.

The screenshot shows the 'Control Files' page in Oracle Enterprise Manager. It has tabs for 'General', 'Advanced', and 'Record Section'. A red box highlights the 'Backup To Trace' button in the top right corner. Below it, there's a section titled 'Control File Mirror Images' with a note about having at least two control files on separate disks. A table lists two control files: 'VALID current.260.689752023' and 'VALID current.256.689752023', with their file paths: '+DATA/orcl/controlfile/' and '+FRA/orcl/controlfile/'. Another red box highlights the 'General' tab of a secondary panel titled 'Control Files' which displays 'Control File Information'.

모든 콘트롤 파일이 손실될 경우 콘트롤 파일 Trace 백업을 사용하여 recovery할 수 있습니다.

Control File Information

Control files store the status of the database physical structure. It is crucial to database operation.

Database ID 1217532758
Control File Type CURRENT
Control File Creation Date June 17, 2009 5:47:05 AM
Control File Sequence Number 1557
Last Change Number 1025918
Date Last Modified June 18, 2009 12:13:30 PM
Control File AutoBackup Enabled [Click here to disable](#)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

콘트롤 파일을 Trace File로 백업

데이터베이스의 콘트롤 파일을 관리하려면 Enterprise Manager > Server > Control Files를 선택합니다. 콘트롤 파일에는 추가 백업 옵션이 있으며 이를 통해 콘트롤 파일을 Trace file로 백업할 수 있습니다. 콘트롤 파일 Trace 백업에는 모든 콘트롤 파일이 손실될 경우 콘트롤 파일을 재생성하는 데 필요한 SQL 문이 포함되어 있습니다.

드문 일이기는 하지만 올바로 구성된 데이터베이스(콘트롤 파일의 여러 복사본이 개별 디스크와 개별 컨트롤러에 놓임)에서 모든 콘트롤 파일이 동시에 손실되는 경우가 있습니다. 따라서 관리자는 데이터베이스의 물리적 구조가 변경될 때마다(테이블스페이스나 데이터 파일을 추가하거나 다른 리도 로그 그룹을 추가하는 등) 콘트롤 파일을 Trace file로 백업해야 합니다.

콘트롤 파일의 Trace 복사본은 Enterprise Manager(슬라이드 참조) 또는 다음 SQL 명령을 사용하여 생성할 수 있습니다.

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

Trace 백업이 DIAGNOSTIC_DEST 초기화 파라미터로 지정한 위치에 생성됩니다. 예를 들어, orcl_vktm_8400.trc와 같은 파일 이름으로 /u01/app/oracle/diag/rdbms/orcl/orcl/trace 위치에 생성됩니다.

Control Files 페이지의 Advanced 탭에서는 콘트롤 파일에 포함된 정보를 볼 수 있습니다.

백업 관리

The screenshot shows the 'Manage Current Backups' interface. At the top, there are four buttons: 'Catalog Additional Files', 'Crosscheck All', 'Delete All Obsolete', and 'Delete All Expired'. Below these are two tabs: 'Backup Sets' (selected) and 'Image Copies'. A 'Search' section follows, with dropdowns for 'Status' (set to 'Available'), 'Completion Time' (set to 'Within a month'), and a 'Go' button. Under 'Contents', several checkboxes are checked: Datafile, Archived Redo Log, SPFILE, and Control File. The main area is titled 'Results' and contains a table of backup details:

Select	Key	Tag	Completion Time	Contents	Device Type	Status	Keep	Pieces
<input type="checkbox"/>	4	TAG20090618T121325	Jun 18, 2009 12:13:27 PM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	3	BACKUP_ORCL.ORACLE_061809120854	Jun 18, 2009 12:13:21 PM	ARCHIVED LOG	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	2	TAG20090618T121228	Jun 18, 2009 12:12:56 PM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	1	BACKUP_ORCL.ORACLE_061809120854	Jun 18, 2009 12:12:20 PM	DATAFILE	DISK	AVAILABLE	NO	1

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

백업 관리

기존 백업을 관리하려면 Enterprise Manager > Availability > Manage Current Backup을 선택합니다. 이 페이지에서는 백업이 완료된 시기, 백업이 생성된 위치(디스크 또는 테이프) 및 백업 사용 가능 여부를 확인할 수 있습니다.

Manage Current Backups 페이지의 맨 위에는 기존 백업에 사용할 수 있는 다음과 같은 네 개의 버튼이 있습니다.

- Catalog Additional Files:** Enterprise Manager를 통해 작동하는 RMAN이 권장되는 백업 생성 방법이지만 RMAN이 인식하지 못하는 결과가 포함된 어떤 다른 환경에서 다른 방법으로 생성된 Image Copy나 백업 셋이 있을 수 있습니다. 이 작업은 이러한 파일을 식별하여 카탈로그에 추가합니다.
- Crosscheck All:** RMAN이 불필요한 백업을 자동으로 삭제할 수 있지만 유저 역시 운영 체제 명령을 사용하여 이러한 백업을 삭제할 수 있습니다. RMAN을 사용하지 않고 백업을 삭제하는 경우 카탈로그는 유저가 카탈로그와 실제 내용을 상호 확인하기 전까지 백업이 삭제되었는지 여부를 알 수 없습니다.
- Delete All Obsolete:** Retention 정책보다 오래된 백업을 삭제합니다.
- Delete All Expired:** 상호 확인이 수행될 때 발견되지 않은 백업에 대한 카탈로그 리스트를 삭제합니다.

백업 보고서 보기

View Backup Report
The following backup jobs are known to the database. The data is retrieved from the database control file.

Search
Status All Start Time Within 1 month Type All Go

Results
Total 1 (Completed ✓ 1)

Backup Name	Status	Start Time	Time Taken	Type	Output Devices	Input Size	Output Size	Output Rate (Per Sec)
BACKUP_ORCL.ORACLE_061809120854	COMPLETED	Jun 18, 2009 12:09:16	00:04:14	DB FULL	DISK	1.64G	1.34G	5.42M

Related Links
Manage Current

Inputs

Datafile Number	Output Type	Output Key	File Size	Tablespace	Checkpoint Time	Incremental Level	Compression Ratio	Corrupted Blocks	File Creation Time	File Checkpoint SCN	Resetlogs SCN
1	BACKUPSET	1	750.01M	SYSTEM	Jun 18, 2009 12:09:18 PM GMT+07:00		1.158	0	Apr 24, 2009 10:31:11 AM GMT+07:00	1025302	740137
2	BACKUPSET	1	651.26M	SYSAUX	Jun 18, 2009 12:09:18 PM GMT+07:00		1.43	0	Apr 24, 2009 10:31:17 AM GMT+07:00	1025302	740137
3	BACKUPSET	1	100.01M	UNDOTBS1	Jun 18, 2009 12:09:18 PM GMT+07:00		11.595	0	Apr 24, 2009 11:29:42 AM GMT+07:00	1025302	740137
4	BACKUPSET	1	148.20M	USERS	Jun 18, 2009 12:09:18 PM GMT+07:00		1.102	0	Apr 24, 2009 10:31:30 AM GMT+07:00	1025302	740137
5	BACKUPSET	1	100.01M	EXAMPLE	Jun 18, 2009 12:09:18 PM GMT+07:00		1.442	0	Jun 17, 2009 5:49:29 AM GMT+07:00	1025302	740137

Control Files

Output Type	Output Key	Checkpoint Time	File Size	File Checkpoint SCN	Resetlogs SCN
BACKUPSET		2 Jun 18, 2009 12:12:28 PM GMT+07:00	9.30M	1025706	740137
BACKUPSET		4 Jun 18, 2009 12:13:25 PM GMT+07:00	9.30M	1025894	740137

SPFile

Backup Set	Modificaton Time	File Size
	4 Jun 18, 2009 11:21:10 AM GMT+07:00	0.00K
	2 Jun 18, 2009 11:21:10 AM GMT+07:00	0.00K

Archived Logs

Output Type	Output Key	Thread Number	Sequence Number	File Size	Low Time	High Time	Compression Ratio	Resetlogs SCN
BACKUPSET	3	1	14	42.14M	Jun 18, 2009 6:41:27 AM GMT+07:00	Jun 18, 2009 12:13:04 PM GMT+07:00	1	740137

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

백업 보고서 보기

백업 작업에 대한 정보는 Enterprise Manager > Availability > Backup Reports를 선택해도 볼 수 있습니다. 이러한 정보의 내용은 콘트롤 파일을 기반으로 합니다. 백업 보고서에는 요약 정보는 물론 타이밍, SCN, 크기 조정, 압축, 손상(있는 경우) 등과 같은 특정 작업의 입력 및 출력에 대한 세부 정보도 포함됩니다.

Fast Recovery Area 모니터

Flash Recovery

This database is using a flash recovery area. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location

Flash Recovery Area Size MB

Flash Recovery Area Size must be set when the location is set.

Non-reclaimable Flash Recovery Area (GB) **1.5**

Reclaimable Flash Recovery Area (MB) **53**

Free Flash Recovery Area (GB) **2.8**

Enable Flashback Database*
Flashback database can be used for fast database point-in-time recovery, as it returns the database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate. The flash recovery area must be set to enable flashback database.

Flashback Retention Time Hours

Current size of the flashback logs(GB) n/a

Lowest SCN in the flashback data n/a

Flashback Time n/a

Apply initialization parameter changes to SPFILE only. If not checked, parameter changes will be made to both the SPFILE and the running instance.

* Changes to this setting or parameter require a database restart.

Flash Recovery Area Usage

Category	Size (GB)	Percentage
Backup Piece	1.34	30.8%
Online Log	0.15	3.4%
Control File	0.01	0.2%
Archived Redo Log	0	0%
Image Copy	0	0%
Flashback Log	0	0%
Usable	2.85	65.6%

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Fast Recovery Area 모니터

아카이브된 로그를 이 위치에 기록하도록 구성한 경우에는 반드시 이 공간을 모니터하여 해당 용량에 도달하지 않는지 확인해야 합니다. Instance가 공간 부족으로 인해 아카이브된 로그를 생성할 수 없는 경우 관리자가 이러한 상황을 해결할 때까지 Instance가 일시 중지됩니다.

Enterprise Manager > Availability > Recovery Settings를 선택합니다. 이 페이지에서 다음을 수행할 수 있습니다.

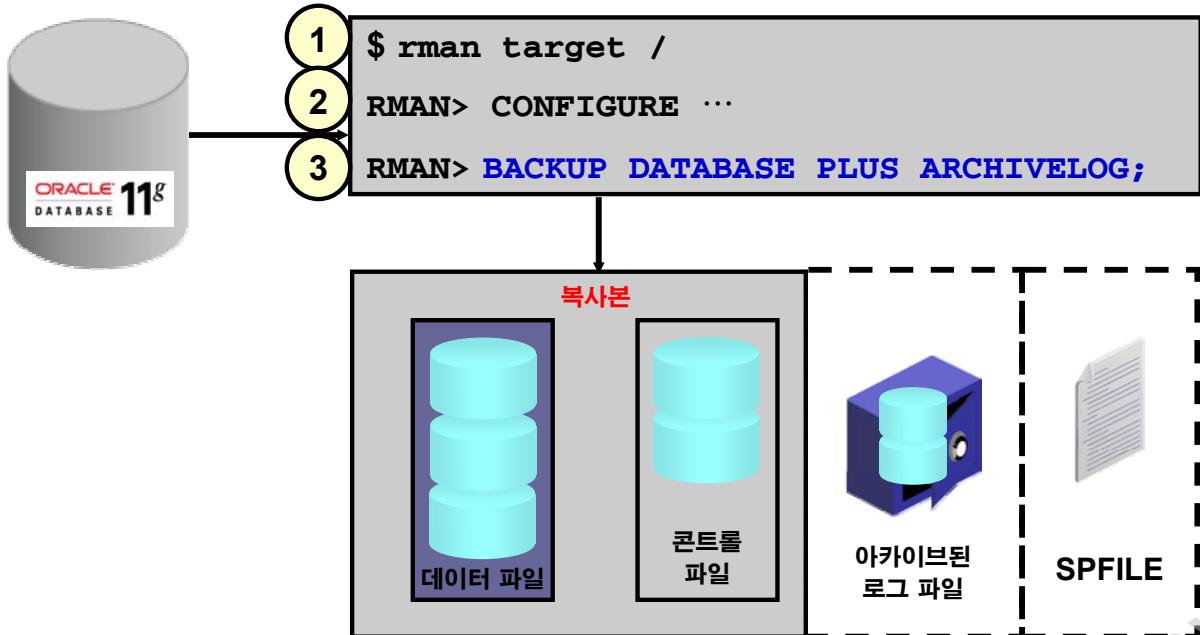
- Fast Recovery Area가 소비한 공간 크기 확인
- Fast Recovery Area의 위치 지정
- Fast Recovery Area의 크기 지정
- Flashback Database 구성
- Retention 시간 지정

파일이 불필요해지는 시기, 즉 파일이 데이터 Recovery 목표를 충족하는 데 더 이상 필요가 없는 상황은 Retention 시간에 따라 결정됩니다. 오라클 데이터베이스는 이 저장 영역을 자동으로 관리하여 더 이상 필요 없는 파일을 삭제합니다. Recovery 영역을 백업할 때 Fast Recovery Area에서 아카이브된 리두 로그가 순상되거나 액세스가 불가능한 경우 RMAN이 다른 아카이브된 리두 로그 대상에 Failover할 수 있습니다.

백업을 테이프에 정기적으로 복사하면 Fast Recovery Area에 다른 파일을 위한 공간이 마련되지만 테이프에서 파일을 검색하면 데이터베이스 복원 및 Recovery 시간이 길어지게 됩니다.

주: Flash Recovery Area는 Fast Recovery Area로 이름이 변경되었지만 Enterprise Manager 인터페이스는 아직 이러한 이름 변경이 반영되도록 개선되지 않았습니다.

RMAN 명령행 사용



ORACLE

Copyright © 2010, Oracle. All rights reserved.

RMAN 명령행 사용

1. 터미널 세션에서 RMAN을 시작하고 Target Database에 연결합니다.
2. 다음 구성 명령을 실행합니다.
 - CONFIGURE DEFAULT DEVICE TYPE TO disk;
 - CONFIGURE DEVICE TYPE DISK BACKUP TYPE TO COPY;
 - CONFIGURE CONTROLFILE AUTOBACKUP ON;
3. Whole 데이터베이스 백업은 모든 데이터 파일 및 콘트롤 파일의 복사본입니다. 선택적으로 서버 파라미터 파일(SPFILE) 및 아카이브된 리드 로그 파일을 포함할 수 있습니다. RMAN을 사용하여 모든 데이터베이스 파일의 Image Copy를 만들려면 데이터베이스를 마운트하거나 열고 RMAN을 시작한 다음 슬라이드에 표시된 BACKUP 명령을 입력하기만 하면 됩니다. 선택적으로 아카이브 로그 파일을 백업할 때는 DELETE INPUT 옵션을 입력할 수 있습니다. 이 옵션을 입력하면 아카이브 로그 파일이 백업된 후에 제거됩니다. 이는 공간이 부족해지면 파일을 삭제하는 공간 관리 작업을 자동으로 수행하는 Fast Recovery Area를 사용하지 않는 경우에 특히 유용합니다. 이 경우 슬라이드의 명령은 다음과 같습니다.

RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;

다음 명령을 사용하여 데이터베이스의 모든 데이터 파일과 콘트롤 파일의 이전 Image Copy의 백업(백업 셋 또는 Image Copy)을 생성할 수도 있습니다.

RMAN> BACKUP COPY OF DATABASE;

퀴즈

RMAN으로 수행되는 Image Copy 백업은 변경 사항 추적 기능을 사용하여 마지막 백업 이후 변경되지 않은 블록을 건너뛸 수 있습니다.

1. 참
2. 거짓

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 일관성 있는 데이터베이스 백업 생성
- 데이터베이스를 종료하지 않고 백업
- Incremental 백업 생성
- 데이터베이스 백업 자동화
- 백업 관리 및 백업 보고서 보기
- Fast Recovery Area 모니터

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

연습 15 개요: 데이터베이스 백업 생성

이 연습에서는 다음 내용을 다룹니다.

- 유저 작업을 위해 데이터베이스가 열려 있는 상태로 데이터베이스 백업
- 데이터베이스에 대한 야간의 자동 Incremental 백업 일정 잡기

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 Recovery 수행

Copyright © 2010, Oracle. All rights reserved.

ORACLE®

목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- Recovery 수행 필요성 판단
- 다양한 인터페이스 액세스(예: Enterprise Manager 및 명령행)
- RMAN(Recovery Manager) 및 Data Recovery Advisor 같은 사용 가능한 옵션 설명 및 사용
- Recovery 수행:
 - 콘트롤 파일
 - 리두 로그 파일
 - 데이터 파일

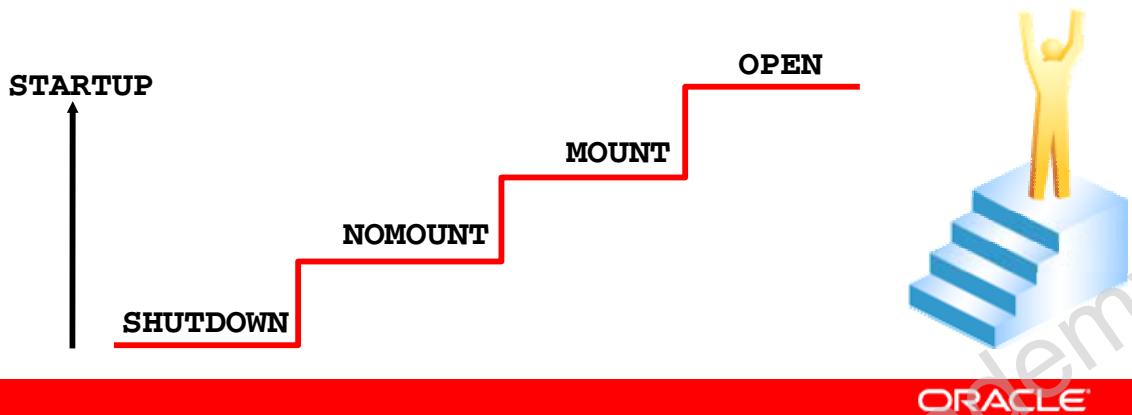
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 열기

데이터베이스를 열려면 다음 조건을 갖춰야 합니다.

- 모든 콘트롤 파일이 존재하고 동기화되어야 합니다.
- 모든 온라인 데이터 파일이 존재하고 동기화되어야 합니다.
- 리두 로그 그룹마다 멤버가 하나 이상 있어야 합니다.



Copyright © 2010, Oracle. All rights reserved.

데이터베이스 열기

데이터베이스가 종료 단계에서 완전히 열린 단계로 이동하면 다음과 같은 단계에서 내부 일관성 검사를 수행합니다.

- NOMOUNT**: Instance가 NOMOUNT(STARTED라고도 함) 상태에 도달하려면 Instance가 초기화 파라미터 파일을 읽어야 합니다. Instance가 NOMOUNT 상태로 진입하는 동안에는 데이터베이스 파일을 검사하지 않습니다.
- MOUNT**: Instance가 MOUNT 상태로 이동하면 초기화 파라미터 파일에 나열된 모든 콘트롤 파일이 존재하고 동기화되었는지 여부를 검사합니다. 콘트롤 파일이 하나라도 누락되었거나 손상된 경우 Instance는 콘트롤 파일 누락을 알리는 오류를 관리자에게 반환하고 NOMOUNT 상태로 유지됩니다.
- OPEN**: Instance가 MOUNT 상태에서 OPEN 상태로 이동하면 다음을 수행합니다.
 - 콘트롤 파일이 인식한 모든 리두 로그 그룹에 멤버가 하나 이상 있는지 여부를 검사합니다. 누락된 멤버는 Alert log에 기록됩니다.

데이터베이스 열기(계속)

- 콘트롤 파일이 인식한 모든 데이터 파일(오프라인이 아닌 데이터 파일)이 존재하는지 확인합니다. 오프라인 파일은 관리자가 온라인으로 가져올 때까지 검사하지 않습니다. 데이터 파일이 SYSTEM 또는 UNDO 테이블스페이스에 속하지 않는 경우 관리자는 데이터 파일을 오프라인으로 만들고 Instance를 열 수 있습니다. 누락된 파일이 있는 경우 첫번째 누락 파일을 알리는 오류가 관리자에게 반환되며 Instance는 MOUNT 상태로 유지됩니다. Instance가 누락된 파일을 찾을 때 문제를 일으킨 첫번째 파일만 오류 메시지에 표시됩니다. Recovery가 필요한 모든 파일을 찾으려면 관리자는 다음과 같이 v\$recover_file dynamic performance 뷰를 확인하여 주의가 필요한 전체 파일 리스트를 가져올 수 있습니다.

```
SQL> startup
ORACLE instance started.
Total System Global Area  171966464 bytes
  Fixed Size                  775608  bytes
  Variable Size                145762888  bytes
  Database Buffers            25165824  bytes
  Redo Buffers                 262144  bytes
Database mounted.

ORA-01157: cannot identify/lock data file 4 - see DBWR trace
file
ORA-01110: data file 4: '/oracle/oradata/orcl/users01.dbf'
SQL> SELECT name, error
  2  FROM v$datafile
  3  JOIN v$recover_file
  4  USING (file#);
NAME                           ERROR
-----
/oracle/oradata/orcl/users01.dbf    FILE NOT FOUND
/oracle/oradata/orcl/example01.dbf  FILE NOT FOUND
```

- 오프라인이나 읽기 전용이 아닌 모든 데이터 파일이 콘트롤 파일과 동기화되어 있는지 확인합니다. 필요한 경우 Instance Recovery가 자동으로 수행됩니다. 하지만 파일이 온라인 리두로그 그룹을 사용하여 recovery 할 수 없을 정도로 동기화 범위를 벗어난 경우 관리자는 Media Recovery를 수행해야 합니다. Media Recovery가 필요한 파일이 있는 경우 recovery가 필요한 첫번째 파일을 알리는 오류 메시지가 관리자에게 반환되며 Instance는 MOUNT 상태로 유지됩니다.

```
ORA-01113: file 4 needs media recovery
ORA-01110: data file 4: '/oracle/oradata/orcl/users01.dbf'
다시 v$recover_file이 주의가 필요한 전체 파일 리스트를 제공합니다.
현재 존재하며 Media Recovery가 필요한 파일이 나열되지만 오류 메시지는
표시되지 않습니다.
```

데이터베이스를 열린 상태로 유지

데이터베이스가 열린 후 다음 항목이 없는 경우 데이터베이스가 실패합니다.

- 콘트롤 파일
- 시스템 또는 언두 테이블스페이스에 속하는 데이터 파일
- 전체 리두 로그 그룹
(사용 가능한 그룹 멤버가 하나라도 있으면 Instance는 열린 상태로 유지됨)

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터베이스를 열린 상태로 유지

데이터베이스가 열린 후에는 콘트롤 파일 손실, 전체 리두 로그 그룹 손실, SYSTEM 또는 UNDO 테이블스페이스에 속하는 데이터 파일의 손실 등과 같은 Media Failure로 인해 Instance Failure 가 발생할 수 있습니다. 비활성 리두 로그 그룹이 손실된 경우 로그 스위치로 인해 데이터베이스가 결국 실패합니다.

대부분의 경우 실패한 Instance가 완전히 종료되지는 않지만 작업을 계속 수행할 수는 없습니다. 이러한 유형의 Media Failure로부터의 Recovery는 데이터베이스가 다운된 상태에서 수행해야 합니다. 따라서 관리자는 Recovery를 수행하기 전에 SHUTDOWN ABORT 명령을 사용해야 합니다.

다른 테이블스페이스에 속하는 데이터 파일이 손실된 경우에는 Instance failure가 발생하지 않으므로 다른 테이블스페이스에서 작업을 진행하도록 데이터베이스를 연 상태에서 recovery 할 수 있습니다.

이러한 오류는 Alert Log File을 검사하거나 Data Recovery Advisor를 사용하여 감지할 수 있습니다.

Data Recovery Advisor

- 빠른 Failure 감지, 분석 및 복구
- 다운타임 및 런타임 Failure
- 유저에 대한 방해 최소화
- 유저 인터페이스:
 - Enterprise Manager GUI(여러 경로)
 - RMAN 명령행
- 지원되는 데이터베이스 구성:
 - 단일 Instance
 - RAC 지원 불가
 - Standby Database로의 Failover는 지원하지만 Standby Database의 분석 및 복구는 지원하지 않음



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Recovery Advisor

Data Recovery Advisor는 오류가 발생할 때 Data Failure 정보를 자동으로 수집합니다. Failure를 사전(Proactive) 검사할 수도 있습니다. 이 모드에서 Data Recovery Advisor는 데이터베이스 프로세스가 손상을 발견하여 오류를 알리기 전에 Data Failure를 감지 및 분석할 수 있습니다. 복구 작업은 항상 수동으로 제어됩니다.

Data Failure는 매우 심각한 문제가 될 수 있습니다. 예를 들어, 현재 로그 파일이 없는 경우에는 데이터베이스를 열 수 없습니다. 데이터 파일의 블록 손상 등과 같은 일부 Data Failure는 데이터베이스가 다운되도록 하거나 오라클 데이터베이스를 열 수 없게 하지는 않기 때문에 그리 심각한 문제는 아닙니다. Data Recovery Advisor는 필수적인 데이터베이스 파일이 없거나, 일관되지 않거나, 손상되었기 때문에 데이터베이스를 시작할 수 없는 경우와 런타임에 파일 손상이 발견되는 경우를 모두 처리합니다.

심각한 Data Failure를 해결하기 위해 주로 사용되는 방법은 다음과 같습니다.

1. Data Guard 구성인 경우 Standby Database로 Failover합니다. 그러면 유저가 최대한 빨리 온라인으로 돌아올 수 있습니다.
2. Data Failure의 주 원인을 복구합니다(유저에게는 영향을 주지 않음).

Data Recovery Advisor의 기능(계속)

유저 인터페이스

Data Recovery Advisor는 EM(Enterprise Manager) Database Control 및 Grid Control에서 사용할 수 있습니다. Failure가 발생하면 여러 가지 방법으로 Data Recovery Advisor에 액세스할 수 있습니다. 다음 예제는 모두 Database Instance 홈 페이지에서 시작됩니다.

- Availability 탭 페이지 > Perform Recovery > Advise and Recover
- Active Incidents 링크 > Support Workbench "Problems" 페이지의 Checker Findings 탭 페이지 > Launch Recovery Advisor
- Database Instance Health > Incidents 섹션에서 특정 링크(예: ORA 1578) 누르기 > Support Workbench, Problems Detail 페이지 > Data Recovery Advisor
- Database Instance Health > Related Links 섹션: Support Workbench > Checker Findings 탭 페이지: Launch Recovery Advisor
- Related Link: Advisor Central > Advisors 탭 페이지: Data Recovery Advisor
- Related Link: Advisor Central > Checkers 탭 페이지: Details > Run Detail 탭 페이지: Launch Recovery Advisor

또한 RMAN 명령행을 통해 Data Recovery Advisor를 사용할 수도 있습니다.

```
rman target /  
rman> list failure all;
```

지원되는 데이터베이스 구성

현재 릴리스에서는 Data Recovery Advisor가 단일 Instance 데이터베이스를 지원합니다. Oracle Real Application Clusters 데이터베이스는 지원되지 않습니다.

Data Recovery Advisor는 Standby Database에서 전송된 블록 또는 파일을 사용하여 Primary Database의 Failure를 복구할 수 없습니다. 또한 Data Recovery Advisor를 사용하여 Standby Database의 Failure를 진단하고 복구할 수는 없습니다. 그러나 앞에서 언급했듯이 Data Recovery Advisor는 Standby Database로의 Failover를 복구 옵션으로 지원합니다.

콘트롤 파일 손실

콘트롤 파일이 손실 또는 손상되면 Instance가 정상적으로 중지됩니다.

- 콘트롤 파일이 ASM 디스크 그룹에 저장된 경우 Recovery 옵션은 다음과 같습니다.
 - Enterprise Manager를 사용하여 단계별 Recovery를 수행합니다.
 - 데이터베이스를 NOMOUNT 모드로 설정하고 RMAN 명령을 사용하여 기존 콘트롤 파일에서 콘트롤 파일을 복원합니다.

```
RMAN> restore controlfile from  
'+DATA/orcl/controlfile/current.260.695209463';
```

- 콘트롤 파일이 일반 파일 시스템 파일로 저장된 경우에는 다음을 수행합니다.
 - 데이터베이스를 종료합니다.
 - 손실된 콘트롤 파일을 대체할 기존 콘트롤 파일을 복사합니다.

콘트롤 파일이 성공적으로 복원된 다음에는 데이터베이스를 엽니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

콘트롤 파일 손실

콘트롤 파일 손실을 Recovery하기 위한 옵션은 콘트롤 파일의 저장 영역 구성 및 적어도 하나의 콘트롤 파일이 남아 있는지 또는 모두 손실되었는지에 따라 달라집니다.

ASM 저장 영역을 사용 중이고 적어도 하나의 콘트롤 파일 복사본이 남아 있는 경우 Enterprise Manager를 사용하여 단계별 Recovery를 수행하거나 다음과 같이 RMAN을 사용하여 수동 Recovery를 수행할 수 있습니다.

- 데이터베이스를 NOMOUNT 모드로 설정합니다.
- RMAN에 연결하고 `restore controlfile` 명령을 실행하여 기존 콘트롤 파일에서 콘트롤 파일을 복원합니다. 예:
`restore controlfile from
'+DATA/orcl/controlfile/current.260.695209463';`
- 콘트롤 파일이 성공적으로 복원된 다음 데이터베이스를 엽니다.

콘트롤 파일이 일반 파일 시스템 파일로 저장되었고 적어도 하나의 콘트롤 파일 복사본이 남아 있는 경우, 데이터베이스가 작동 중지된 상태에서 남은 콘트롤 파일 중 하나를 손실된 파일의 위치에 바로 복사하면 됩니다. Media failure가 디스크 드라이버 또는 컨트롤러의 손실로 인해 발생한 경우 남아 있는 콘트롤 파일 중 하나를 다른 위치로 복사하고 Instance의 파라미터 파일이 새 위치를 가리키도록 갱신합니다. 또는 초기화 파라미터 파일에서 누락된 콘트롤 파일에 대한 참조를 삭제할 수 있습니다. Oracle은 동시에 적어도 두 개 이상의 콘트롤 파일을 보유할 것을 권장합니다.

주: 모든 콘트롤 파일이 손실된 경우의 Recovery 방법은 Oracle Database 11g: Administration Workshop II 과정에서 다릅니다.

리두 로그 파일 손실

리두 로그 파일 그룹의 멤버가 손실되었지만 아직 그룹에 멤버가 하나 이상 있는 경우 다음과 같은 결과가 발생합니다.

- 정상적인 Instance 작업에는 영향을 주지 않습니다.
- Alert log에 멤버를 찾을 수 없음을 알리는 메시지가 전송됩니다.
- 손실된 리두 로그 멤버를 삭제하고 새 멤버를 추가하여 누락된 로그 파일을 복원할 수 있습니다.
- 누락된 로그 파일이 있는 그룹이 아카이브된 경우 로그 그룹을 지워서 누락된 파일을 재생성할 수 있습니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

리두 로그 파일 손실

단일 리두 로그 그룹 멤버를 손실에서 Recovery하는 것은 실행 중인 Instance에 영향을 미치지 않습니다.

이 Recovery를 수행하려면 다음과 같이 하십시오.

1. Alert log를 검토하여 누락된 로그 파일이 있는지 확인합니다.
2. 먼저 손실된 리두 로그 멤버를 삭제하여 누락된 파일을 복원합니다.

SQL> ALTER DATABASE DROP LOGFILE MEMBER

'+DATA/orcl/onlinelog/group_1.261.691672257' ;

그린 다음 새 멤버를 추가하여 손실된 리두 로그 멤버를 교체합니다.

SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA' TO GROUP 2;

Enterprise Manager을 사용하여 로그 파일 멤버를 삭제하고 재생성 할 수도 있습니다.

주: 리두 로그 파일에 대해 OMF를 사용 중이고 위 구문을 사용하여 기존 그룹에 새 리두 로그 멤버를 추가하면, 이 새 리두 로그 멤버 파일은 OMF 파일이 되지 않습니다. 새 리두 로그 멤버가 OMF 파일이 되도록 해야 하는 경우의 가장 쉬운 Recovery 옵션은 새 리두 로그 그룹을 생성하고 누락된 리두 로그 멤버가 있는 리두 로그 그룹을 삭제하는 것입니다.

리두 로그 파일 손실(계속)

3. Media failure가 디스크 드라이버 또는 컨트롤러의 손실로 인해 발생한 경우 누락된 파일의 이름을 바꿉니다.
4. 그룹이 이미 아카이브되었거나 유저가 NOARCHIVELOG 모드인 경우 로그 그룹을 지우고 누락된 파일을 다시 생성하여 문제를 해결할 수 있습니다. 적절한 그룹을 선택하고 Clear Logfile 작업을 선택합니다. 또는 다음 명령을 사용하여 영향을 받은 그룹을 수동으로 지울 수 있습니다.

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP #
```

주: Database Control에서는 아카이브되지 않은 로그 그룹을 지울 수 없습니다. 지우게 되면 리두 정보 체인이 손상됩니다. 아카이브되지 않은 로그 그룹을 지워야만 하는 경우에는 Whole 데이터베이스를 즉시 전체 백업을 해야 합니다. 이 작업 중 Failure가 일어나면 다른 Failure가 발생할 때 데이터가 손실될 수 있습니다. 아카이브되지 않은 로그 그룹을 지우려면 다음 명령을 사용합니다.

```
SQL> ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP #;
```

NOARCHIVELOG 모드에서의 데이터 파일 손실

데이터베이스가 NOARCHIVELOG 모드이고 데이터 파일이 손실된 경우 다음 작업을 수행하십시오.

1. Instance가 아직 종료되지 않은 경우 Instance를 종료합니다.
2. 모든 데이터 파일 및 콘트롤 파일을 포함하여 전체 데이터베이스를 백업에서 복원합니다.
3. 데이터베이스를 엽니다.
4. 마지막 백업 이후 변경된 모든 사항을 유저가 재입력하게 합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

NOARCHIVELOG 모드에서의 데이터 파일 손실

NOARCHIVELOG 모드의 데이터베이스에서 임의의 데이터 파일이 손실된 경우 콘트롤 파일 및 모든 데이터 파일을 포함하여 데이터베이스를 완전히 복원해야 합니다.

데이터베이스를 NOARCHIVELOG 모드로 설정하면 마지막 백업 시점까지만 Recovery 가 가능합니다. 따라서 유저가 백업 이후 수행된 모든 변경 사항을 재입력해야 합니다.

이러한 유형의 Recovery를 수행하려면 다음과 같이 하십시오.

1. Instance가 아직 종료되지 않은 경우 Instance를 종료합니다.
2. Maintenance 속성 페이지에서 Perform Recovery를 누릅니다.
3. Recovery 유형으로 Whole Database를 선택합니다.

데이터베이스가 Incremental 백업 전략이 있는 NOARCHIVELOG 모드인 경우 먼저 RMAN 이 가장 최근 레벨 0을 복원한 다음 RMAN Recovery가 Incremental 백업을 적용합니다.

ARCHIVELOG 모드에서의 Noncritical 데이터 파일 손실

데이터 파일이 손실되거나 손상되었으며 해당 파일이 SYSTEM 또는 UNDO 테이블스페이스에 속하지 않는 경우 누락된 데이터 파일을 복원하고 Recovery합니다.

Object Level Recovery

Object Type Datafiles

Operation Type Recover to current time
Datafile will be restored as required.
 Restore datafiles
Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.
 Recover from previously restored datafiles
 Block Recovery



ORACLE

Copyright © 2010, Oracle. All rights reserved.

ARCHIVELOG 모드에서의 Noncritical 데이터 파일 손실

데이터베이스가 ARCHIVELOG 모드인 경우 SYSTEM 또는 UNDO 테이블스페이스에 속하지 않는 데이터 파일이 손실되면 누락된 파일의 객체에만 영향을 미칩니다. 데이터베이스의 나머지 부분은 유저가 계속 작업할 수 있는 상태로 유지됩니다.

누락된 데이터 파일을 복원하고 Recovery하려면 다음과 같이 하십시오.

1. Maintenance 속성 페이지에서 Perform Recovery를 누릅니다.
2. Recovery 유형으로 Datafiles를 선택한 다음 "Restore to current time"을 선택합니다.
3. Recovery가 필요한 모든 데이터 파일을 추가합니다.
4. 파일을 기본 위치로 복원할 것인지 또는 새 위치(디스크나 컨트롤러가 누락된 경우)로 복원할 것인지를 결정합니다.
5. RMAN 작업을 제출(submit)하여 누락된 파일을 복원하고 Recovery합니다.

데이터베이스가 ARCHIVELOG 모드이므로 마지막 커밋 시점까지 Recovery가 가능하며 유저가 데이터를 재입력할 필요가 없습니다.

ARCHIVELOG 모드에서의 시스템

Critical 데이터 파일 손실

데이터 파일이 손실되거나 손상되었으며 해당 파일이 SYSTEM 또는 UNDO 테이블스페이스에 속하는 경우 다음 작업을 수행하십시오.

- 1. Instance가 자동으로 종료되거나 종료되지 않을 수 있습니다.
자동으로 종료되지 않으면 SHUTDOWN ABORT를 사용하여
Instance를 종료합니다.**
- 2. 데이터베이스를 마운트합니다.**
- 3. 누락된 데이터 파일을 복원하고 Recovery합니다.**
- 4. 데이터베이스를 엽니다.**



ORACLE

Copyright © 2010, Oracle. All rights reserved.

ARCHIVELOG 모드에서의 시스템 Critical 데이터 파일 손실

SYSTEM 테이블스페이스에 속하거나 UNDO 데이터를 포함하는 데이터 파일은 시스템 Critical 파일로 간주됩니다. 이러한 파일 중 하나가 손실되면 데이터베이스를 연 상태에서 복원할 수 있는 다른 데이터 파일과 달리 MOUNT 상태에서 데이터베이스를 복원해야 합니다.

이 Recovery를 수행하려면 다음과 같이 하십시오.

- 1. Instance가 아직 종료되지 않은 경우 Instance를 종료합니다.**
- 2. 데이터베이스를 마운트합니다.**
- 3. Maintenance 속성 페이지에서 Perform Recovery를 누릅니다.**
- 4. Recovery 유형으로 Datafiles를 선택한 다음 "Restore to current time"을 선택합니다.**
- 5. Recovery가 필요한 모든 데이터 파일을 추가합니다.**
- 6. 파일을 기본 위치로 복원할 것인지 또는 새 위치(디스크나 컨트롤러가 누락된 경우)로 복원할 것인지를 결정합니다.**
- 7. RMAN 작업을 제출(Submit)하여 누락된 파일을 복원하고 Recovery합니다.**
- 8. 데이터베이스를 엽니다. Recovery가 마지막 커밋 시점까지 진행되므로 유저가 데이터를 재입력할 필요가 없습니다.**

Data Failure: 예제



- 액세스할 수 없는 구성 요소: OS 레벨의 누락된 데이터 파일, 잘못된 액세스 권한, 오프라인 테이블스페이스
- 물리적 손상: 블록 체크섬 Failure, 유효하지 않은 블록 헤더 필드 값
- 논리적 손상: 일관되지 않은 딕셔너리, 손상된 행 피스(piece), 손상된 인덱스 항목 또는 손상된 트랜잭션
- 불일치: 데이터 파일 및 온라인 리두 로그보다 오래되거나 최신 버전인 콘트롤 파일
- I/O Failure: 열린 파일 수 제한 초과, 액세스 불가능한 채널, 네트워크 또는 I/O 오류

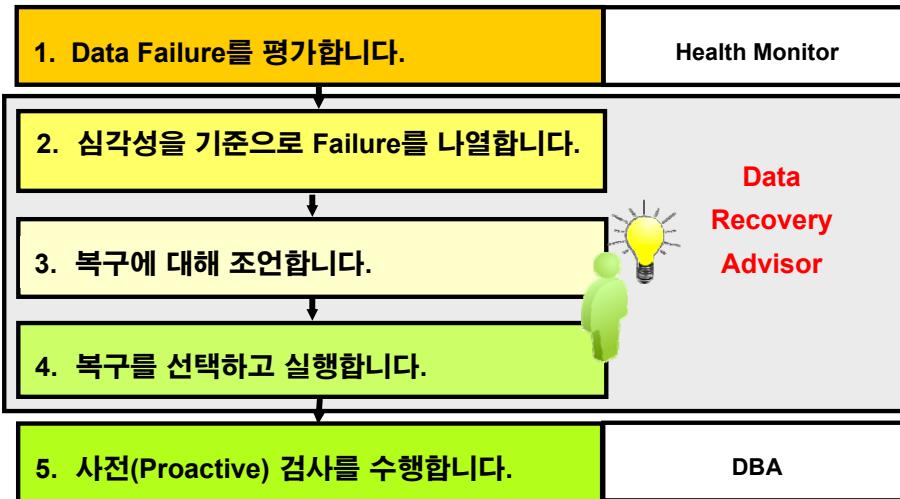
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Data Failure: 예제

Data Recovery Advisor는 Failure를 분석하고 늘어나는 문제 리스트에 대해 복구 옵션을 제안할 수 있습니다.

Data Recovery Advisor



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Recovery Advisor

Oracle Database 11g의 자동 진단 워크플로우는 워크플로우 단계를 자동으로 수행합니다.

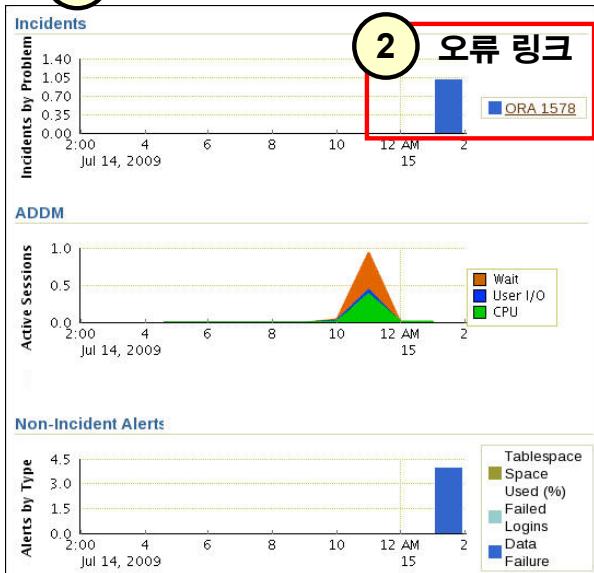
Data Recovery Advisor를 사용하면 조언 및 복구를 시작하기만 하면 됩니다.

1. Health Monitor가 검사를 자동으로 실행하고 Failure 및 현상을 ADR(Automatic Diagnostic Repository)에 "조사 결과"로 기록합니다.
2. Data Recovery Advisor는 조사 결과를 Failure에 통합하고, 이전에 실행한 평가의 결과를 Failure 심각성(Critical 또는 High)과 함께 나열합니다.
3. Failure에 대한 복구 관련 조언을 요청하면 Data Recovery Advisor가 Failure를 자동 및 수동 복구 옵션에 매핑하고, 기본 실행 가능성을 검사하고, 복구 관련 조언을 제시합니다.
4. 복구를 수동으로 실행하거나 Data Recovery Advisor에 실행을 요청할 수 있습니다.
5. Health Monitor 및 Data Recovery Advisor의 자동 기본 "사후(Reactive)" 검사 외에도 오라클은 VALIDATE 명령을 "사전(Proactive)" 검사로 사용할 것을 권장합니다.

Data Failure 액세스

1

데이터베이스 Instance 상태



2 오류 링크

3

문제 상세 정보

Problem Details: ORA 1578

Page Refreshed July 15, 2009 1:36:09 AM GMT+07:00 [Refresh]

Summary

SR#	--	[Edit]
Bug#	--	[Edit]
Active	Yes	
Packaged	No	
Number of Incidents	1	

Last Dumped Incident

Timestamp	July 15, 2009 1:30:50 AM GMT+07:00
Incident Source	System Generated
Impact	
Checkers Run	1
Checker Findings	1

Investigate and Resolve

- Go to My Oracle Support
- Quick Package
- Self Service
- Oracle Support

Assess Damage

- Checker Findings
- Run Checkers
- Database Instance Health

Diagnose

- Alert Log
- Related Problems Across Topology
- Diagnostics for Last Dumped Incident
- Go to My Oracle Support and Research

Resolve

- SQL Repair Advisor
- Data Recovery Advisor

Incidents **Activity Log**

Status: Open Incidents Data Dumped: Yes Go

Select All | Select None | Show All Details | Hide All Details

Select Details	ID	Description	Data Dumped	Active	Status	Timestamp
▶ Show	18345	ORA-1578 [9] [129]	Yes	Yes	Ready	July 15, 2009 1:30:50 AM GMT+07:00

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Failure 액세스

슬라이드의 예제에서는 Health Monitor와 Data Recovery Advisor의 상호 작용을 확인하는 몇 가지 방법 중 하나를 보여줍니다.

Data Failure

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The top navigation bar includes 'Help' and 'Database'. The main content area has a green header bar with 'Information' and two items: 'Database Failures - 1' and 'Current Status - MOUNTED'. Below this is a section titled 'Perform Recovery'.

Oracle Advised Recovery: A message states: 'The Data Recovery Advisor has detected failures. Click on "Advise and Recover" to have Oracle analyze and produce recovery advice.' A red box highlights the 'Advise and Recover' button.

Failures Detected: Critical: 1, High: 0, Low: 0. Failure: System datafile 1: '+DATA/orcl/datafile' Description: '/system.256.692202091' is missing.

User Directed Recovery: Recovery Scope is set to 'Whole Database'. The 'Recover' button is also highlighted with a red box.

Operation Type: The radio button for 'Recover to the current time or a previous point-in-time' is selected. It specifies: 'Datafiles will be restored from the latest usable backup as required.' Other options include 'Restore all datafiles' (Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.) and 'Recover from previously restored datafiles'.

Overview: A list of recovery actions:

- Recover database failures as advised by Oracle
- Restore and/or recover the entire database or selected objects
- Restore files to a new location
- Recover tablespaces to a point-in-time based on a timestamp, system change number (SCN), or log sequence number
- Recover datafile data blocks that are marked as corrupted, or based on datafile block IDs or tablespace block addresses
- Flashback database or tables to a specific system change number (SCN) or timestamp

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Failure

Data Failure는 데이터베이스 또는 해당 구성 요소의 상태를 평가하는 진단 절차인 검사를 통해 발견됩니다. 각 검사는 하나 이상의 Failure를 진단하고 복구에 매핑할 수 있습니다.

검사는 사후(Reactive) 또는 사전(Proactive)일 수 있습니다. 데이터베이스에서 오류가 발생할 때는 사후 검사가 자동으로 실행됩니다. 예를 들어, VALIDATE DATABASE 명령을 실행하여 사전 검사를 시작할 수도 있습니다.

데이터베이스가 "다운" 또는 "마운트" 상태인 경우에는 Enterprise Manager에서 Availability > Perform Recovery를 선택하거나 Perform Recovery 버튼을 누르십시오. Enterprise Manager가 분석을 시작하고 Recovery 조언을 제공하도록 하려면 "Advise and Recover"를 누릅니다.

Data Failure 나열

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The title bar reads "ORACLE Enterprise Manager 11g Database Control". The main menu bar includes "Database", "Help", and "Database". The URL is "Database Instance: orcl.oracle.com > View and Manage Failures". The timestamp is "Last Refresh July 15, 2009 2:38:38 AM GMT+07:00". Below the header, there is a search bar with dropdowns for "Failure Description", "Impact", "Priority", "Status", and "Time Detected". A "Go" button is also present. A red box highlights the "Advise" button in the toolbar above the failure list. The failure list table has columns: "Select Failure Description", "Impact", "Priority", "Status", and "Time Detected". One row is shown, indicating a critical failure where the system datafile is missing. A tip message at the bottom left says: "TIP All CRITICAL failures must be selected before "Advise". All CRITICAL failures must be unselected before "Set Priority High" or "Set Priority Low".". There is also a checkbox for "Use a Recovery Catalog" and a link to "Edit Recovery Catalog Information".

ORACLE

Copyright © 2010, Oracle. All rights reserved.

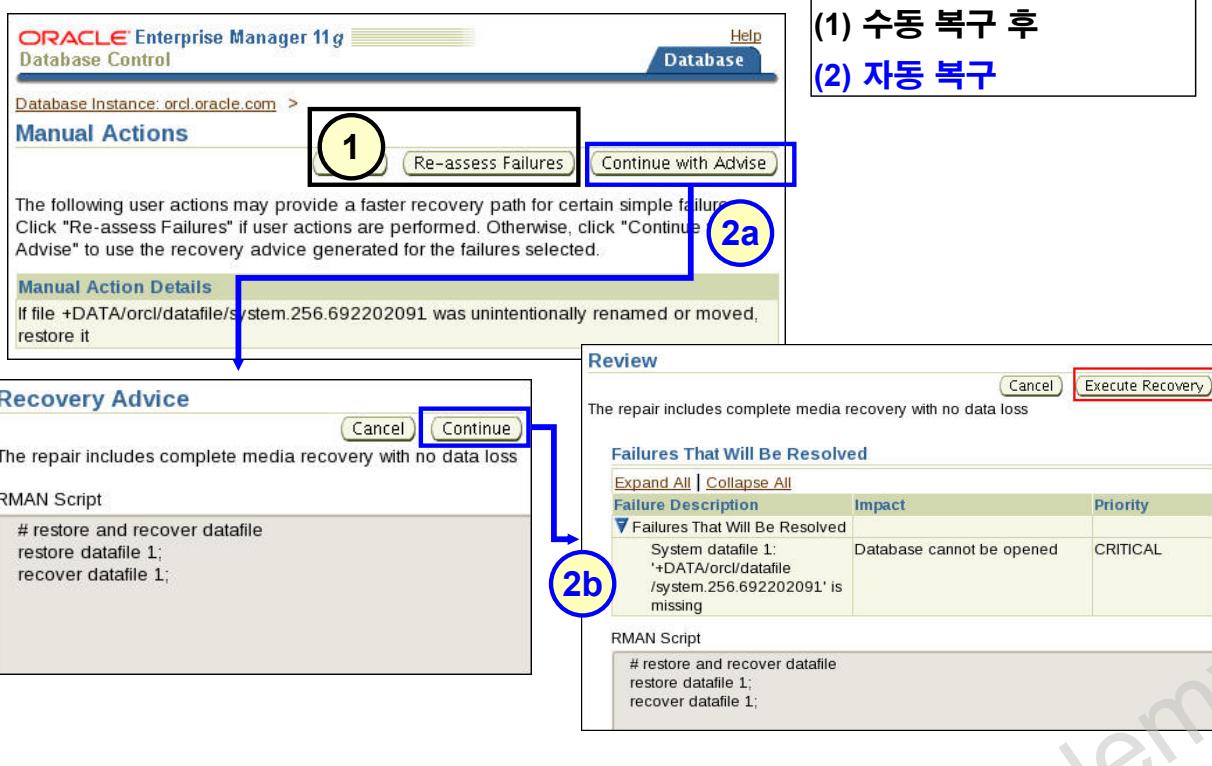
Data Failure 나열

"View and Manage Failures" 페이지가 Data Recovery Advisor의 홈 페이지입니다. 스크린샷의 예제는 Data Recovery Advisor에서 Data Failure 및 상세 정보를 나열하는 방법을 보여줍니다. 시작할 수 있는 작업에는 조언, 우선 순위 설정, Failure 닫기가 있습니다.

기본 LIST FAILURE 명령을 사용하여 Data Failure 및 세부 정보를 표시할 수도 있습니다. Failure 평가는 여기서 시작되지 않으며, 이미 실행되어서 ADR에 저장되어 있습니다.

Failure는 높은 우선 순위에서 낮은 우선 순위의 순서로 나열됩니다(CRITICAL, HIGH, LOW). 동일한 우선 순위의 Failure는 이론 시간에서 늦은 시간의 시간 기록 순으로 나열됩니다.

복구에 대한 조언



ORACLE

Copyright © 2010, Oracle. All rights reserved.

복구에 대한 조언

"View and Manage Failures" 페이지에서 Advise 버튼을 누르면 Data Recovery Advisor 가 수동 확인 목록을 생성합니다. 두 가지 유형의 Failure가 나타날 수 있습니다.

- 수동 작업이 필요한 Failure: 디스크 케이블이 꽂혀 있지 않은 연결 기능 Failure가 한 예입니다.
- 오류가 있는 이전 작업을 언두할 경우 더 빨리 복구되는 Failure: 예를 들어, 데이터 파일의 이름을 실수로 변경한 경우에는 백업에서 RMAN 복원을 시작하는 것보다 이름을 이전 이름으로 다시 변경하는 것이 더 빠릅니다.

다음 작업을 시작할 수 있습니다.

- 수동 복구를 수행한 후에 "Re-assess Failures"를 누릅니다. 해결된 Failure는 암시적으로 닫히고, 남은 Failure가 "View and Manage Failures" 페이지에 표시됩니다.
- "Continue with Advise"를 눌러 자동화된 복구를 시작합니다. Data Recovery Advisor 는 자동화된 복구 옵션을 생성할 때 RMAN이 Failure를 복구하려는 방법을 보여주는 스크립트도 함께 생성합니다. 자동화된 복구를 실행하려면 Continue를 누릅니다. Data Recovery Advisor가 Failure를 자동으로 복구하지 않도록 하려면 이 스크립트를 수동 복구의 시작점으로 사용하면 됩니다.

복구 실행

Database Instance: orcl.oracle.com >

Recovery Succeeded

Recovery succeeded. See Recovery Results below.

Recovery Results

Open Database OK

Recovery Manager: Release 11.2.0.1.0 - Production on Wed Jul 15 02:47:57 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

RMAN>
connected to target database: ORCL (DBID=1219972082, not open)
using target database control file instead of recovery catalog

RMAN> echo set on

media recovery complete, elapsed time **00:00:40**
Finished recover at 15-JUL-09
repair failure complete

RMAN> exit;

Recovery Manager complete.

Open Database OK

The database has been opened successfully.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

복구 실행

Data Recovery Advisor가 이러한 페이지를 표시합니다. 예제에서 성공적인 복구는 40초 이내에 완료됩니다.

Data Recovery Advisor 뷰

다음과 같은 Dynamic 데이터 딕셔너리 뷰를 query할 수 있습니다.

- **V\$IR_FAILURE:** 달한 Failure를 비롯한 모든 Failure 나열
(LIST FAILURE 명령의 결과)
- **V\$IR_MANUAL_CHECKLIST:** ADVISE FAILURE 명령을 수행하면 제공되는 수동 조언 나열
- **V\$IR_REPAIR:** ADVISE FAILURE 명령을 실행하면 수행되는 복구 나열
- **V\$IR_FAILURE_SET:** Failure 및 조언 식별자의 상호 참조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Recovery Advisor 뷰

사용법 예

2007년 6월 21일에 감지된 Failure를 모두 표시해야 한다고 가정합니다.

```
SELECT * FROM v$ir_failure
WHERE trunc (time_detected) = '21-JUN-2007' ;
```

Data Recovery Advisor가 사용하는 Dynamic 데이터 딕셔너리 뷰에 대한 자세한 사항은 *Oracle Database Reference*를 참조하십시오.

퀴즈

다음과 같은 이벤트가 발생할 경우 Instance가 실패하지 않습니다.

1. 남은 다중화된 콘트롤 파일이 있는 경우 콘트롤 파일의 손실
2. SYSTEM 테이블스페이스 손실
3. 손실된 멤버의 동일 그룹에 남은 다중화된 리두 로그
멤버가 있는 경우 리두 로그 멤버 하나의 손실
4. 활성 언두 테이블스페이스 손실

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 3

퀴즈

Data Recovery Advisor에서 사용되는 정보는 Enterprise Manager 인터페이스를 통해서만 제공됩니다.

- 1. 참**
- 2. 거짓**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 2

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- Recovery 수행 필요성 판단
- 다양한 인터페이스 액세스(예: Enterprise Manager 및 명령행)
- RMAN(Recovery Manager) 및 Data Recovery Advisor 같은 사용 가능한 옵션 설명 및 사용
- Recovery 수행:
 - 콘트롤 파일
 - 리두 로그 파일
 - 데이터 파일

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

연습 16 개요: 데이터베이스 Recovery 수행

이 연습에서는 다음 파일이 손실되는 경우의 Recovery 방법을 다룹니다.

- 콘트롤 파일
- Noncritical 데이터 파일
- Critical 시스템 데이터 파일

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

17

데이터 이동

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

목표

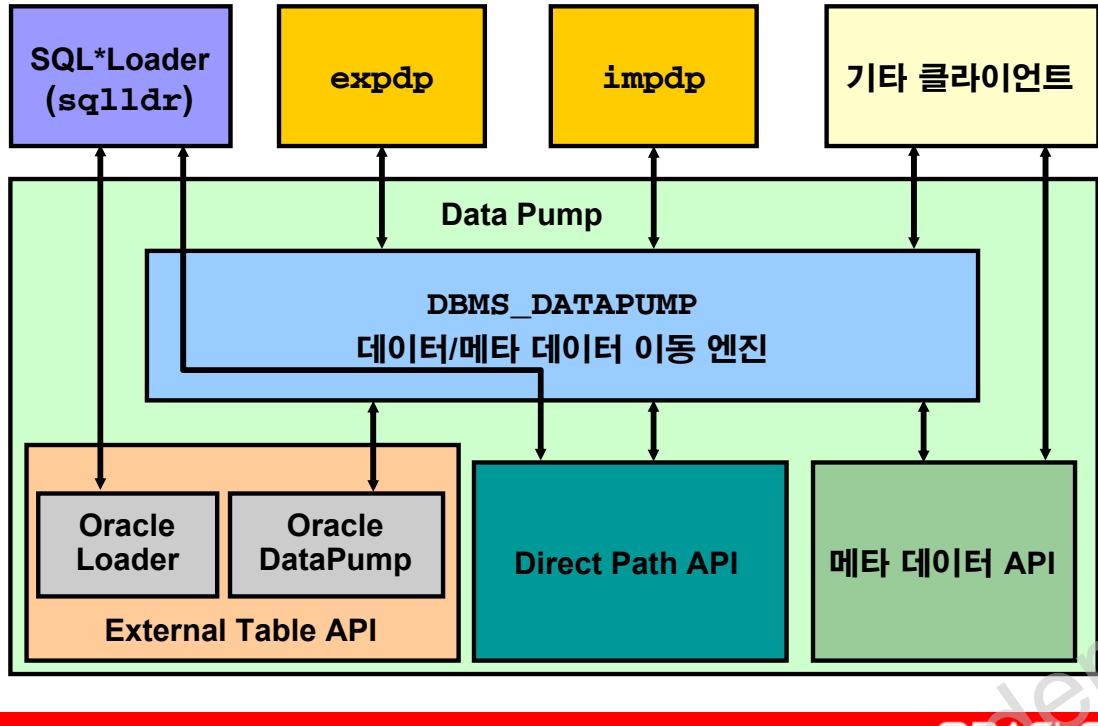
이 단원을 마치면 다음을 수행할 수 있습니다.

- 데이터 이동 방법 설명
- 디렉토리 객체 생성 및 사용
- SQL*Loader를 사용하여 비오라클 데이터베이스 또는 유저 파일에서 데이터 로드
- External Table을 사용하여 플랫폼 독립적인 파일을 통해 데이터 이동
- Oracle Data Pump의 일반적 구조 설명
- Data Pump 엑스포트/임포트를 사용하여 오라클 데이터베이스 간에 데이터 이동

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터 이동: 일반적 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터 이동: 일반적 구조

주요 작동 구성 요소는 다음과 같습니다.

- **DBMS_DATAPUMP:** 대량의 데이터/메타 데이터 이동을 위한 고속 엑스포트 및 임포트 유ти리티용 API를 포함합니다.
- **Direct Path API(DPAPI):** Oracle Database 11g는 로드 취소 및 로드 시 데이터 변환 및 구문 분석을 최소화하는 Direct Path API 인터페이스를 지원합니다.
- **DBMS_METADATA:** 모든 메타 데이터 로드 및 로드 취소를 위해 작업자 프로세스에서 사용됩니다. 데이터베이스 객체 정의는 SQL이 아니라 XML을 사용하여 저장됩니다.
- **External Table API:** ORACLE_DATAPUMP 및 ORACLE_LOADER 액세스 드라이버를 사용하여 External Table에 데이터를 저장할 수 있습니다(플랫폼 독립적인 파일). SELECT 문은 External Table을 오라클 데이터베이스에 저장된 것처럼 읽습니다.
- **SQL*Loader:** External Table과 통합되었으므로 External Table 액세스 파라미터에 로더 콘트롤 파일의 자동 이전 기능을 제공합니다.
- **expdp 및 impdp:** Data Pump 작업을 시작 및 모니터하기 위해 DBMS_DATAPUMP 패키지를 호출하는 Thin layer입니다.
- **기타 클라이언트:** 이 Infrastructure를 활용하는 응용 프로그램(Database Control, 복제, Transportable 테이블스페이스, 유저 응용 프로그램 등)이 여기에 해당됩니다. 또한 SQL*Plus는 진행 중인 작업에 대한 간단한 상태 query를 위해 DBMS_DATAPUMP의 클라이언트로 사용될 수 있습니다.

Oracle Data Pump: 개요

Oracle Data Pump는 고속 데이터/메타 데이터 이동을 위한 서버 기반 기능으로 다음 특징을 갖습니다.

- DBMS_DATAPUMP를 통해 호출할 수 있습니다.
- 다음과 같은 도구를 제공합니다.
 - expdp
 - impdp
 - 웹 기반 인터페이스
- 네 가지 데이터 이동 방법 제공:
 - 데이터 파일 복사
 - Direct path
 - External Table
 - 네트워크 링크 지원
- 장시간 실행되는 작업에서 분리했다가 재연결
- Data Pump 작업 재시작



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Data Pump: 개요

Oracle Data Pump를 사용하면 오라클 데이터베이스에서 데이터/메타 데이터를 매우 빠른 속도로 로드/로드 취소할 수 있습니다. Data Pump Infrastructure는 PL/SQL 패키지 DBMS_DATAPUMP를 통해 호출할 수 있습니다. 그런 다음 Data Pump를 사용하여 커스텀 데이터 이동 유ти리티를 구축할 수 있습니다.

Oracle Database 11g는 다음과 같은 도구를 제공합니다.

- 명령행 엑스포트 및 임포트 클라이언트(각각 expdp 및 impdp)
- Database Control에서 액세스할 수 있는 웹 기반 엑스포트/임포트 인터페이스

Data Pump는 Direct path 또는 External Table 중 사용할 데이터 액세스 방식을 자동으로 결정합니다. Data Pump는 테이블의 구조가 Direct path를 허용하고 최대의 단일 스트림 성능이 요구되는 경우 Direct path 로드/로드 취소를 사용합니다. 그러나 클러스터화된 테이블, 참조 무결성 제약 조건, 암호화된(encrypted) 열 또는 기타 다양한 항목이 있는 경우 Data Pump는 Direct path 대신 External Table을 사용하여 데이터를 이동합니다.

Data Pump에는 장시간 실행되는 작업에서 작업에 영향을 주지 않고 분리했다가 재연결할 수 있는 기능이 있어서 작업 실행 중 여러 위치에서 작업을 모니터할 수 있습니다. 중단된 모든 Data Pump 작업은 메타 정보가 그대로 남아 있는 한 데이터 손실 없이 재시작될 수 있습니다. 이것은 작업이 자발적으로 중단되었든 시스템 장애로 인해 비자발적으로 중단되었든 관계 없습니다.

Oracle Data Pump: 이점

Data Pump는 이전 데이터 이동 도구에 비해 다음과 같은 여러 이점 및 새로운 기능을 제공합니다.

- 하위단계(Fine-grained) 객체 및 데이터 선택
- 데이터베이스 버전의 명시적 지정
- 병렬 실행
- 엑스포트 작업 공간 소비량 예측
- 분산 환경의 네트워크 모드
- 재매핑 기능
- 데이터 샘플링 및 메타 데이터 압축
- Data Pump 엑스포트 중 데이터 압축
- 암호화(encryption)를 통한 보안
- XMLType 데이터를 CLOB로 엑스포트하는 기능
- 이전 임포트 및 엑스포트 파일을 지원하는 기존 모드

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Data Pump: 이점

EXCLUDE, INCLUDE 및 CONTENT 파라미터는 하위단계(Fine-grained) 객체 및 데이터 선택에 사용됩니다.

VERSION 파라미터를 사용하여 이동할 객체의 데이터베이스 버전을 지정하면 Data Pump를 지원하는 이전 릴리스의 오라클 데이터베이스와 호환 가능한 덤프 파일 세트를 생성할 수 있습니다.

PARALLEL 파라미터를 사용하여 엑스포트 작업을 대신해 작동하는 활성 실행 서버의 최대 스레드 수를 지정할 수 있습니다.

ESTIMATE_ONLY 파라미터를 사용하면 엑스포트를 실제로 수행하지 않고도 엑스포트 작업이 소비하는 공간의 양을 예측할 수 있습니다.

네트워크 모드에서는 원격 데이터베이스에서 덤프 파일 세트로 바로 엑스포트할 수 있습니다. 이 작업은 소스 시스템에 연결되는 데이터베이스 링크를 사용하여 수행할 수 있습니다.

임포트 작업 중에 대상 데이터 파일 이름, 스키마 및 테이블스페이스를 변경할 수 있습니다.

또한 Data Pump 엑스포트를 수행할 때 소스 데이터베이스에서 샘플링하여 로드 취소할 데이터의 백분율을 지정할 수 있습니다. 이 작업은 SAMPLE 파라미터를 지정하여 수행할 수 있습니다.

COMPRESSION 파라미터를 사용하여 엑스포트 덤프 파일에서 디스크 공간 소모를 줄이도록 메타 데이터를 압축해야 할지 여부를 나타낼 수 있습니다. 메타 데이터를 압축할 경우 임포트 중 자동으로 압축이 풀립니다.

Oracle Database 11g의 향상된 Data Pump 기능

Oracle Database 11g에는 다음을 수행할 수 있도록 하는 새 기능이 추가되었습니다.

- 엑스포트 중에 데이터와 메타 데이터 모두, 데이터만, 메타 데이터만 압축하거나 데이터를 압축하지 않음
- 다음 영역에서 추가 암호화(encryption) 옵션 지정:
 - 엑스포트 중에 데이터와 메타 데이터를 모두 암호화하거나, 데이터 또는 메타 데이터만 암호화하거나, 데이터를 암호화하지 않거나, 암호화된 열만 암호화하도록 선택할 수 있습니다.
 - 엑스포트 중에 특정 암호화 알고리즘을 사용하도록 지정할 수 있습니다.
 - 엑스포트 중에 암호화 및 암호 해독(decryption)에 사용할 보안 유형을 지정할 수 있습니다. 예를 들어, 덤프 파일 세트를 다른 데이터베이스나 원격 데이터베이스로 임포트할 때 전송 중에 보안 상태를 유지해야 합니다. 또는 Oracle Encryption Wallet을 사용하여 온사이트에 덤프 파일 세트를 임포트하거나, Oracle Encryption Wallet을 사용할 수 없는 오프사이트에 덤프 파일 세트를 임포트해야 할 수도 있습니다.
- Transportable 방식을 사용하여 테이블 모드 엑스포트 및 임포트를 수행하고 임포트 작업 중에 Partition 테이블을 처리하는 방법 지정
- 엑스포트 작업 중에 기존 덤프 파일 결쳐쓰기
- 임포트 작업 중에 테이블 이름 바꾸기
- Nondeferred 제약 조건 위반이 발견되는 경우에도 데이터 로드를 진행하도록 지정(External Table 액세스 방식을 사용하는 임포트 작업에만 해당)
- XMLType 열이 해당 열에 대해 정의된 XMLType 저장 형식에 관계없이 압축되지 않은 CLOB 형식으로 엑스포트되도록 지정
- 엑스포트 중에 지정된 열의 원래 값을 소스로 사용하는 재매핑 함수를 지정하고 덤프 파일에서 원래 값을 대체하는 재매핑된 값 반환
- 새 데이터베이스로 임포트되는 데이터 재매핑
- 원본 엑스포트(exp) 및 임포트(imp) 스크립트 사용을 지원하기 위한 기존 모드

Data Pump를 위한 디렉토리 객체

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The top navigation bar includes links for Setup, Preferences, Help, Logout, and Database. It also indicates that the user is logged in as SYS. The main content area is titled "Directory Objects". A search bar at the top allows users to search by object name. Below the search bar is a note explaining the search behavior. A "Selection Mode" dropdown is set to "Single". A toolbar below the mode dropdown includes buttons for Edit, View, Delete, Actions, Create Like, and Go. A table lists various directory objects with their names and paths. The table has two columns: "Name" and "Path". The "Name" column contains entries like DATA_FILE_DIR, DATA_PUMP_DIR, LOG_FILE_DIR, MEDIA_DIR, ORACLE_OCM_CONFIG_DIR, SS_OE_XMLDIR, SUBDIR, and XMLDIR. The "Path" column shows the corresponding file system locations. At the bottom of the table are buttons for Database, Setup, Preferences, Help, and Logout.

Select	Name	Path
<input checked="" type="radio"/>	DATA_FILE_DIR	/u01/app/oracle/product/11.2.0/dbhome_1/demo/schema/sales_history/
<input type="radio"/>	DATA_PUMP_DIR	/u01/app/oracle/admin/orcl/dpdump/
<input type="radio"/>	LOG_FILE_DIR	/u01/app/oracle/product/11.2.0/dbhome_1/demo/schema/log/
<input type="radio"/>	MEDIA_DIR	/u01/app/oracle/product/11.2.0/dbhome_1/demo/schema/product_media/
<input type="radio"/>	ORACLE_OCM_CONFIG_DIR	/u01/app/oracle/product/11.2.0/dbhome_1/ocr/state
<input type="radio"/>	SS_OE_XMLDIR	/u01/app/oracle/product/11.2.0/dbhome_1/demo/schema/order_entry/
<input type="radio"/>	SUBDIR	/u01/app/oracle/product/11.2.0/dbhome_1/demo/schema/order_entry//2002/Sep
<input type="radio"/>	XMLDIR	/ade/b/1191423112/oracle/rdbms/xml

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump를 위한 디렉토리 객체

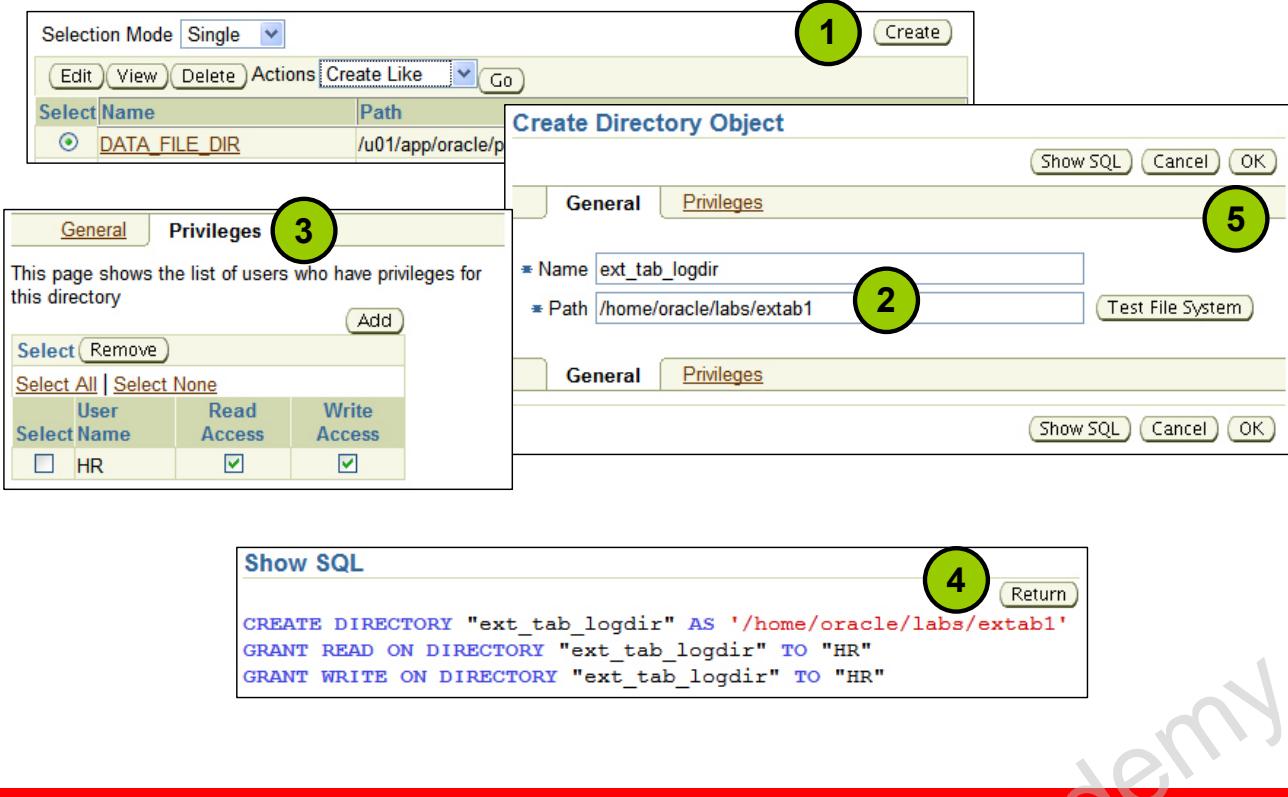
디렉토리 객체는 서버의 파일 시스템에 있는 물리적 디렉토리를 나타내는 논리적 구조이며 특정 운영 체제 디렉토리의 위치를 포함합니다. 이 디렉토리 객체 이름은 Enterprise Manager에서 사용될 수 있으므로 디렉토리 경로 지정을 하드 코딩할 필요가 없습니다. 따라서 더욱 효율적으로 파일을 관리할 수 있습니다. 디렉토리 객체는 SYS 유저가 소유합니다. 모든 디렉토리가 단일 이름 공간(즉, SYS)에 위치하기 때문에 디렉토리 이름은 데이터베이스에서 고유합니다.

Data Pump는 클라이언트가 아니라 서버에 있는 파일을 액세스하기 때문에 Data Pump에 대해 파일 위치를 지정할 때 디렉토리 객체가 필요합니다.

Enterprise Manager에서 Schema > Database Objects > Directory Objects를 선택합니다.

디렉토리 객체를 편집하거나 삭제하려면 객체를 선택하고 해당 버튼을 누릅니다.

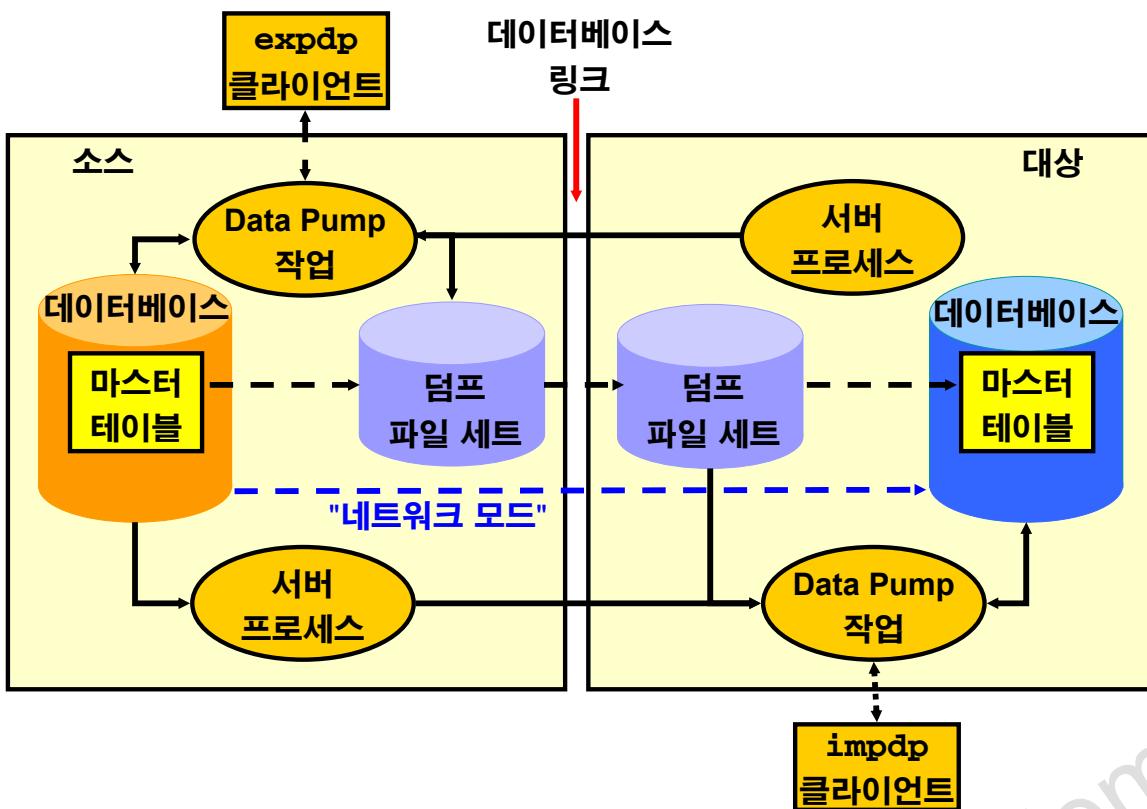
디렉토리 객체 생성



디렉토리 객체 생성

1. Directory Objects 페이지에서 Create 버튼을 누릅니다.
2. 디렉토리 객체의 이름과 디렉토리 객체를 매핑할 OS 경로를 입력합니다. OS 디렉토리를 사용하기 전에 먼저 생성해야 합니다. Test File System 버튼을 눌러 이 기능을 테스트할 수 있습니다. 테스트를 위해 이 OS 디렉토리에 대한 권한을 가진 OS 유저임을 나타내는 호스트 로그인 인증서를 제공합니다.
3. 디렉토리 객체에 대한 권한은 서버 파일 시스템의 물리적 디렉토리에 대한 OS 권한과 동일하지 않습니다. 개별 디렉토리 객체에 대한 유저 권한을 관리할 수 있습니다. 따라서 보안 레벨이 향상되고 이러한 객체를 세부적으로 제어할 수 있습니다. Privileges 페이지에서 Add를 눌러 읽기나 쓰기 권한(또는 둘 다)을 부여할 유저를 선택합니다.
4. Show SQL을 눌러 기본 명령문을 확인합니다. 완료되면 Return을 누릅니다.
5. OK를 눌러 객체를 생성합니다.

Data Pump 엑스포트/임포트 클라이언트: 개요



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 엑스포트/임포트 클라이언트: 개요

Data Pump 엑스포트는 덤프 파일 세트로 알려진 일련의 운영 체제 파일로 데이터/메타 데이터를 로드 취소하는 유ти리티입니다. Data Pump 임포트는 엑스포트 덤프 파일 세트에 저장된 메타 데이터/데이터를 대상 시스템으로 로드하는 데 사용됩니다.

Data Pump API는 클라이언트가 아닌 서버의 파일을 액세스합니다.

또한 이러한 유ти리티를 사용하여 원격 데이터베이스에서 덤프 파일 세트로 바로 엑스포트하거나, 중간에 개입하는 파일 없이 소스 데이터베이스에서 바로 Target Database를 로드할 수 있습니다. 이것을 네트워크 모드라고 합니다. 이 모드는 특히 읽기 전용 소스 데이터베이스의 데이터를 엑스포트할 때 유용합니다.

모든 Data Pump 작업의 중심에는 마스터 테이블이 있습니다. 이것은 Data Pump 작업을 실행하는 유저의 스키마에 생성되는 테이블입니다. MT는 작업의 모든 측면을 유지 관리합니다. MT는 파일 기반 엑스포트 작업 동안 구축되며 마지막 단계로 덤프 파일 세트에 기록됩니다. 반면, MT를 현재 유저의 스키마로 로드하는 것은 파일 기반 임포트 작업의 첫번째 단계이며 임포트되는 모든 객체를 연속적으로 생성하는 데 사용됩니다.

주: MT는 계획되거나 계획되지 않은 작업이 중단될 경우 Data Pump를 재시작할 수 있는 수단입니다. MT는 Data Pump 작업이 정상적으로 완료되면 삭제됩니다.

Data Pump 유ти리티: 인터페이스 및 모드

- Data Pump 엑스포트/임포트 인터페이스:
 - 명령행
 - 파라미터 파일
 - 대화식 명령행
 - Enterprise Manager
- Data Pump 엑스포트/임포트 모드:
 - 전체
 - 스키마
 - 테이블
 - 테이블스페이스
 - Transportable 테이블스페이스



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 유ти리티: 인터페이스 및 모드

다음 인터페이스 중 하나를 사용하여 Data Pump 엑스포트/임포트와 상호 작용할 수 있습니다.

- 명령행 인터페이스: 명령행에 직접 대부분의 엑스포트 파라미터를 지정할 수 있습니다.
- 파라미터 파일 인터페이스: 파라미터 파일에 모든 명령행 파라미터를 지정할 수 있습니다. 단, PARFILE 파라미터는 지정할 수 없습니다.
- 대화식 명령 인터페이스: 터미널에 대한 로깅이 정지되고 다양한 명령을 입력할 수 있는 엑스포트 또는 임포트 프롬프트를 표시합니다. 명령행 인터페이스 또는 파라미터 파일 인터페이스로 시작한 엑스포트 작업 도중 [Ctrl] + [C]를 누르면 이 모드가 활성화됩니다. 대화식 명령 모드는 또한 실행 중인 작업 또는 정지된 작업에 연결할 때에도 활성화됩니다.
- 웹 인터페이스: Database Control 홈 페이지에서 Data Movement 탭을 누른 다음 Move Row Data 영역에서 Export to Files, Import from Files, Import from Database 링크 중 하나를 선택합니다.

Data Pump 엑스포트/임포트는 데이터베이스의 서로 다른 부분을 로드 췌소/로드할 수 있는 다양한 모드를 제공합니다. 각 모드는 적절한 파라미터를 사용하여 명령행에서 지정됩니다.

사용 가능한 모드가 슬라이드에 나열되어 있으며 이는 원래의 엑스포트 및 임포트 유ти리티에서와 동일합니다.

Database Control을 사용하여 Data Pump 엑스포트

The screenshot shows the Oracle Enterprise Manager Database Control interface. At the top, there's a navigation bar with tabs: Home, Performance, Availability, Server, Schema, Data Movement (which is selected), and Software and Support. Below the navigation bar, there are four main sections: Move Row Data, Move Database Files, Streams, and Advanced Replication. The 'Move Row Data' section has a red box around the 'Export to Export Files' link, and a red arrow points from this box to the 'Export: Export Type' dialog box below. The 'Export: Export Type' dialog box shows the following details:

- Database:** orcl.oracle.com
- Export: Export Type:** Schemas (radio button selected)
- Host Credentials:**
 - * Username: oracle
 - * Password: [REDACTED]
 - Save as Preferred Credential
- Buttons:** Cancel, Continue

At the bottom right of the interface is the ORACLE logo, and at the very bottom is the copyright notice: Copyright © 2010, Oracle. All rights reserved.

Database Control을 사용하여 Data Pump

Enterprise Manager Database Control은 Data Pump 엑스포트 및 임포트 프로시저 수행 과정을 안내하는 마법사를 제공합니다. 위 예제에서는 Data Pump 엑스포트를 보여 줍니다.

데이터베이스 Instance home 페이지에서 Data Movement 탭을 누르고 웹 페이지의 "Move Row Data" 섹션 아래에서 Data Pump 엑스포트 및 임포트 옵션을 찾습니다. "Export to Export Files"를 눌러서 Data Pump 엑스포트 세션을 시작합니다.

다음에 표시되는 window는 엑스포트 유형 선택 window입니다. 권한 있는 유저가 데이터베이스 Instance에 연결된 경우 엑스포트 유형에는 다음이 포함됩니다.

- 데이터베이스
- 스키마
- 테이블
- 테이블스페이스

비관리 계정이 사용된 경우 엑스포트 유형 리스트는 다음으로 제한됩니다.

- 스키마
- 테이블

Continue를 눌러 엑스포트를 진행합니다.

Data Pump 엑스포트 예제: 기본 옵션

Export: Options

Database orcl.oracle.com

Maximum Number of Threads in Export Job 1

This option allows you to make tradeoffs between resource consumption and elapsed time. Parallelism is only available in Enterprise Edition.

Estimate Disk Space

Calculates an estimate of how much disk space the export job will consume (in bytes). The estimate is for table row data only and does not include metadata.

Blocks
Estimate will be calculated by multiplying the number of database blocks used by the target objects times the appropriate block sizes. This method will provide the quickest rough estimate.

Statistics
Estimate will be calculated using per-table statistics. This method will provide the most accuracy if all target tables have been recently analyzed.

Optional File

Generate Log File

Directory Object DATA_PUMP_DIR

Log File EXPDAT.LOG

Show Advanced Options

Cancel Finish Step 1 of 4 Next

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 엑스포트 예제: 기본 옵션

Options 페이지에는 Data Pump 엑스포트 작업에 대해 구성 가능한 옵션이 표시됩니다.

"Maximum Number of Threads in Export Job" 항목은 명령행의 PARALLEL 파라미터에 해당합니다. 지정한 값은 덤프 파일 세트에 있는 파일 수보다 작거나 같아야 합니다. 이 옵션은 사용되는 병렬 I/O 서버 프로세스의 수를 결정하지만, 병렬 query 작업에서 query 조정자 역할을 수행하는 마스터 컨트롤 프로세스 및 작업자 프로세스는 이를 합계에 포함하지 않습니다.

엑스포트 작업이 소비하는 예상 디스크 공간(바이트 단위)은 표준 출력 및 로그 파일에 출력됩니다. 예상 값은 블록 수와 블록 크기를 곱한 값 또는 최근 통계 분석을 기반으로 할 수 있습니다. 이 옵션은 ESTIMATE 명령행 파라미터에 해당합니다.

엑스포트 작업에서는 진행 중인 작업, 완료된 작업 및 발생한 오류와 관련된 모든 메시지에 대해 선택적인 로그 파일을 지정할 수 있습니다. 엑스포트를 수행하는 유저는 로그 파일에 대해 지정된 디렉토리 객체에 대한 쓰기 권한이 필요합니다. 지정하려는 로그 파일이 존재하는 경우 겹쳐써집니다. 이 옵션은 명령행의 LOGFILE 파라미터에 해당합니다.

Data Pump 엑스포트 예제: 고급 옵션

Content

What to Export from the Source Database All
Export both metadata and data
 Data Only
Export only table row data
 Metadata Only
Export only database object definitions

Export Content Include All Objects
 Include Only Objects Specified Below
 Exclude Only Objects Specified Below

Objects to Include or Exclude

Select Object Type	Object Name Expression
No items found	
Add Another Row	

Object Name Expression example: "IN('EMP','DEPT)" or, to include every object except those of a particular type not beginning with PRO, select EXCLUDE with an expression of "NOT LIKE 'PRO%'"

Flashback

Export read-consistent view of data
 As the specified System Change Number (SCN)
SCN
 As the SCN which most closely matches the specified time
Date
Time AM PM

Query

Specify SELECT statement predicate clauses to be applied to tables being exported. If a Table Name is not supplied for a particular Predicate Clause, the Predicate Clause is applied to (and must make sense for) all tables being exported.

Select Predicate Clause	Table Name
No items found	
Add	

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 엑스포트 예제: 고급 옵션

"Show Advanced Options" 링크를 누르면 고급 옵션을 표시할 수 있습니다.

페이지의 "Content" 섹션에서는 엑스포트가 업로드하는 항목(데이터만, 메타 데이터만 또는 둘 다)을 필터링할 수 있습니다. 이 옵션은 CONTENT 명령행 파라미터에 해당합니다. "Content" 섹션에서는 또한 INCLUDE 및 EXCLUDE 명령행 파라미터를 지정할 수 있습니다.

EXCLUDE 파라미터는 임의의 데이터베이스 객체 유형을 엑스포트 또는 임포트 작업에서 제외할 수 있습니다. 선택 사항인 이름 수식자를 사용하면 다음 예제와 같이 지정된 각 객체 유형 내에서 더욱 세밀하게 선택할 수 있습니다.

```
EXCLUDE=VIEW  
EXCLUDE=PACKAGE  
EXCLUDE=INDEX: "LIKE 'EMP%' "
```

INCLUDE 파라미터는 지정된 객체 유형과 객체만 작업에 포함시킵니다.

구문: INCLUDE = object_type[: "name_expr"]

Flashback query는 데이터를 엑스포트하는 중에 사용할 수 있습니다.

QUERY 파라미터는 원래의 엑스포트 유ти리티와 비슷한 방식으로 작동하지만 두 가지 측면에서 기능이 향상되었습니다. 첫번째는 테이블 이름으로 조건을 제한하여 해당 테이블에만 적용되도록 하는 것이고, 두번째는 임포트 중에도 사용할 수 있는 것입니다. 예를 들면, 다음과 같습니다.

```
QUERY=hr.employees: "WHERE department_id in (10,20)
```

Data Pump 엑스포트 예제: 파일

Progress: Options (Completed) → **Files** (Completed) → Schedule → Review

Export: Files

Database orcl.oracle.com

Specify the directory object and file name, and maximum size for the export files on the database server machine.

(Remove)

Select Directory Object	File Name	Maximum File Size (MB)
DATA_PUMP_DIR	HR_SCHEMA.DMP	

Add Another Row

You can wildcard a set of dump files using '%U' in the filename. A '%D' wildcard will be replaced with the date the job is run using a YYMMDD format.

Cancel | Finish | Back | Step 2 of 4 | Next

[Database](#) | [Help](#) | [Logout](#)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 엑스포트 예제: 파일

DUMPFILE 파라미터는 디스크 기반 덤프 파일의 이름과 디렉토리(선택 사항)를 지정합니다. 여러 개의 파일을 지정할 경우 쉼표로 분리된 리스트를 사용하거나 별도의 DUMPFILE 파라미터로 지정할 수 있습니다. 파일 이름에는 여러 개의 파일이 생성될 수 있음을 의미하는 %U 치환 변수가 포함될 수 있습니다. %U는 결과 파일 이름에서 고정 너비인 두 개의 문자로 확장되며 01에서 시작하여 정수 단위로 증가합니다. DUMPFILE이 지정되지 않은 경우 기본적으로 expdat.dmp가 사용됩니다. 생성되는 덤프 파일은 기본적으로 자동 확장됩니다. FILESIZE가 지정된 경우 각 파일은 크기가 FILESIZE 바이트이고 확장할 수 없습니다. 더 많은 덤프 공간이 필요하고 %U가 있는 템플리트가 제공된 경우 새 파일은 자동으로 FILESIZE 바이트로 생성되고 그렇지 않을 경우 클라이언트는 새 파일을 추가하라는 메시지를 받습니다.

%U가 있는 템플리트가 지정된 경우 처음에 생성된 파일의 수는 PARALLEL 파라미터와 같습니다. 기존 파일의 이름이 결과 파일 이름과 일치해도 기존 파일을 기본적으로 겹쳐쓰지는 않지만 대신 오류가 발생하며 작업이 중지됩니다. 파일을 겹쳐쓰려는 경우 REUSE_DUMPFILES=Y를 설정합니다. 엑스포트를 계속하려면 Next를 누릅니다.

주: 여러 개의 덤프 파일 템플리트가 제공된 경우 순환 방식으로 덤프 파일을 생성하는데 사용됩니다.

Data Pump 엑스포트 예제: 파일(계속)

- Data Pump 엑스포트를 사용하기 전에 수동으로 디렉토리 객체를 생성할 필요가 없습니다. UNIX 또는 Windows 플랫폼에서 새로 생성되든 스크립트를 통해 간단히 관계없이, 모든 데이터베이스에 대해서는 이름이 DATA_PUMP_DIR인 기본 디렉토리 객체가 생성됩니다. DATA_PUMP_DIR 디렉토리에 대한 액세스 권한은 EXP_FULL_DATABASE 및 IMP_FULL_DATABASE 롤에 자동으로 부여됩니다. DATA_PUMP_DIR 디렉토리는 다음 위치 중 하나에 생성됩니다.
 - <ORACLE_BASE>/admin/DB_UNIQUE_NAME/dpdump
 - <ORACLE_HOME>/admin/DB_UNIQUE_NAME/dpdumpDATA_PUMP_DIR에 대한 정확한 디렉토리 경로 지정은 ORACLE_BASE 및 ORACLE_HOME 시스템 환경 변수 값과 DATA_PUMP_DIR 하위 디렉토리가 있는지 여부에 따라 다릅니다. 대상 시스템에 ORACLE_BASE가 정의된 경우 이 값이 사용됩니다. 그렇지 않은 경우 ORACLE_HOME의 값이 사용됩니다. DATA_PUMP_DIR 서브 디렉토리가 없는 경우에는 다음의 기본 경로가 사용됩니다.
ORACLE_HOME/rdbms/log

주: 어떤 경우든 작업을 수행하려면 디렉토리 객체에 대한 적절한 액세스 권한이 있어야 합니다. 엑스포트의 경우 모든 파일에 대한 쓰기 액세스 권한이 필요하고, 임포트의 경우 덤프 파일에 대한 읽기 액세스 권한과 로그 파일 및 SQL 파일에 대한 쓰기 액세스 권한이 필요합니다.

Data Pump 엑스포트 예제: 일정

Options Files Schedule Review

Export: Schedule

Database **orcl.oracle.com**

Specify a name and description for the export job. Specify a date to start the job.

Job Parameters

Job Name **Export_HR_Schema**
Description **Full export of HR schema**

Job Schedule

Time Zone **(UTC+00:00) Universal Time**

Start

Immediately
 Later

Date **Jul 11, 2009**
(example: Jul 11, 2009)

Time **5 20 AM**

Repeat

One Time Only
 Interval

Frequency **1 Minutes**

Repeat Until

Indefinite
 Custom

Date **Jul 11, 2009**
(example: Jul 11, 2009)

Time **5 30 AM**
(Ignored except when repeating by minutes or hours.)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 엑스포트 예제: 일정

이 마법사를 통해 생성된 Data Pump 작업은 Enterprise Manager Database Control을 사용하여 반복 가능한 작업으로 일정을 잡을 수 있습니다. Job Name을 지정하지 않으면 시스템에서 생성된 이름이 사용됩니다. Next를 눌러 엑스포트를 진행합니다.

Data Pump 엑스포트 예제: 검토

The screenshot shows the 'Export: Review' step of the Oracle Data Pump interface. At the top, there are four tabs: Options, Files, Schedule, and Review (which is highlighted). Below the tabs, the database is set to 'orcl.oracle.com'. On the right, there are 'Cancel', 'Back', 'Step 4 of 4', and a red-bordered 'Submit Job' button. The main area displays the export configuration:

Export Type	Schemas
Statistics type	Estimate optimizer statistics when data is imported
Parallelism	1
Files to Export	DATA_PUMP_DIR HR_SCHEMA.DMP
Log File	DATA_PUMP_DIR EXPDAT.LOG
Job Schedule	Immediately

A 'Hide PL/SQL' link is present. Below it, a large scrollable text area contains the following PL/SQL code:

```
declare
  h1 NUMBER;
begin
  h1 := dbms_datapump.open (operation => 'EXPORT', job_mode => 'SCHEMA', job_name =>
'EXPORT_HR_SCHEMA', version => 'COMPATIBLE');
  dbms_datapump.set_parallel(handle => h1, degree => 1);
  dbms_datapump.add_file(handle => h1, filename => 'EXPDAT.LOG', directory => 'DATA_PUMP_DIR',
filetype => 3);
  dbms_datapump.set_parameter(handle => h1, name => 'KEEP_MASTER', value => 0);
  dbms_datapump.metadata_filter(handle => h1, name => 'SCHEMA_EXPR', value => 'IN("HR")');
```

Below the code, a 'Processing' section indicates the job is in progress:

Submit job is progressing. This may take some time.
This may take several minutes. This page will automatically forward to the next page when done.

A clock icon and the text 'Process is in progress.' are shown. A tip message states: 'TIP This operation cannot be cancelled. It will continue even if the browser window is closed.'

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 엑스포트 예제: 검토

"Review" 페이지에는 입력된 정보에 대한 요약 내용이 표시되고 이 페이지에서는 엑스포트 작업에 사용되는 PL/SQL 프로시저 구문을 볼 수 있습니다. "Submit Job" 버튼을 눌러 계속 진행합니다. 작업이 제출(submit)되면 취소할 수 없으며, 브라우저를 닫는 것이 안전합니다.

Data Pump 임포트 예제: impdp

Data Pump는 명령행에서 호출하여 추가 명령행 옵션을 지정할 수 있습니다.

```
$ impdp hr DIRECTORY=DATA_PUMP_DIR \
DUMPFILE=HR_SCHEMA.DMP \
PARALLEL=1 \
CONTENT=ALL \
TABLES="EMPLOYEES" \
LOGFILE=DATA_PUMP_DIR:import_hr_employees.log \
JOB_NAME=importHR \
TRANSFORM=STORAGE:n
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 임포트 예제: impdp

Data Pump는 임포트 및 엑스포트 작업을 위한 명령행 클라이언트를 제공합니다. 위 예제에서는 impdp 유ти리티를 사용한 Data Pump 임포트를 보여 줍니다. 명령행을 사용하여 Data Pump를 실행하면 더 많은 옵션이 제공됩니다. 전체 옵션 리스트를 보려면 "Oracle Database Utilities 11g Release 2 (11.2)" 설명서를 참조하십시오.

Data Pump 임포트: 변형

다음을 재매핑할 수 있습니다.

- REMAP_DATAFILE을 사용하여 데이터 파일 재매핑
- REMAP_TABLESPACE를 사용하여 테이블스페이스 재매핑
- REMAP_SCHEMA를 사용하여 스키마 재매핑
- REMAP_TABLE을 사용하여 테이블 재매핑
- REMAP_DATA를 사용하여 데이터 재매핑

```
REMAP_TABLE = 'EMPLOYEES' : 'EMP'
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 임포트: 변형

객체 메타 데이터는 덤프 파일 세트에 XML로 저장되므로 임포트 중 DDL이 형성될 때 쉽게 변형을 적용할 수 있습니다. Data Pump 임포트는 다음 몇 가지 변형을 지원합니다.

- REMAP_DATAFILE은 서로 다른 파일 시스템 의미를 사용하는 플랫폼 간에 데이터베이스를 이동할 때 유용합니다.
- REMAP_TABLESPACE를 사용하여 객체를 한 테이블스페이스에서 다른 테이블스페이스로 이동할 수 있습니다.
- REMAP_SCHEMA는 객체 소유권을 변경할 수 있는 이전의 FROMUSER /TOUSER 기능을 제공합니다.
- REMAP_TABLE은 전체 테이블의 이름을 변경할 수 있는 기능을 제공합니다.
- REMAP_DATA는 삽입 중인 데이터를 재매핑할 수 있는 기능을 제공합니다.

Enterprise Manager를 사용하여 Data Pump 작업 모니터

The screenshot shows the Oracle Enterprise Manager interface for managing data movement. The top navigation bar includes links for Home, Performance, Availability, Server, Schema, Data Movement (which is currently selected), and Software and Support. Under the Data Movement section, there are four main categories: Move Row Data, Move Database Files, Streams, and Advanced Replication. Each category has several sub-links. A red box highlights the 'Monitor Export and Import Jobs' link under the Move Row Data category. An arrow points from this link down to a detailed view of an export job.

Export and Import Jobs

Page Refreshed Sep 1, 2008 12:23:20 AM MDT (OK)

In database versions 10g and greater, Enterprise Manager uses data pump jobs to do the actual export and import operations. Although Enterprise Manager exports and imports can also be monitored from their corresponding Job Summary pages, data pump jobs defined outside of Enterprise Manager can only be monitored from here.

(Delete Data Pump Job)

Select Data Pump Job	EM Job	Owner	Job Status
INVENTORY_EXPORT	Yes	DBA1	EXECUTING

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager를 사용하여 Data Pump 작업 모니터

Enterprise Manager 그래픽 유저 인터페이스(GUI)를 사용하여 모든 Data Pump 작업을 모니터할 수 있습니다. 여기에는 expdp 또는 impdp 명령행 인터페이스를 사용하여 생성된 작업이나 DBMS_DATAPUMP 패키지를 사용하여 생성된 작업이 포함됩니다.

현재 작업 상태를 확인하고 EXECUTE, STOP 또는 SUSPEND로 상태를 변경할 수 있습니다.

"Export and Import Jobs" 페이지에 액세스하려면 Maintenance 페이지의 Move Row Data 영역에서 "Monitor Export and Import Jobs" 링크를 누릅니다.

Data Pump 기존 모드를 사용하여 이전

- **imp 및 exp 유ти리티에서 impdp 및 expdp 유ти리티로 변환 지원**
- **기존 모드의 Data Pump:**
 1. 고유한 `imp` 또는 `exp` 파라미터가 발견되면 기존 모드로 전환됩니다.
 2. 이전 구문을 새 구문으로 매핑하려고 시도합니다.
 3. 새 구문을 표시합니다.
 4. 기존 모드를 종료합니다.

최적의 사용법(Best Practice Tip): 오라클은 시간이 허락하는 한 새 구문을 검토하고 스크립트를 변경할 것을 적극 권장합니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 기존 모드를 사용하여 이전

임포트 및 엑스포트 스크립트가 광범위하게 사용되기 때문에 Oracle Database 11g Release 2는 Data Pump 기존 모드에 대한 이전 지원을 제공합니다. Data Pump 유ти리티:

1. 명령행 또는 스크립트에서 `exp/imp`에 고유한 파라미터가 제공된 경우 기존 모드로 전환합니다.
2. 이전 파라미터를 상응하는 `expdp` 또는 `impdp` 파라미터(가능한 경우)로 매핑합니다.
3. 시간이 허락하는 한 유저가 새 구문을 검토하고 스크립트를 변경할 수 있도록 변환된 명령을 표시합니다.
4. 이전 파라미터와 새 파라미터가 혼합되어 제공된 경우 기존 모드를 종료합니다.

Data Pump 파라미터를 원래의 엑스포트 또는 임포트 파라미터와 혼용하면 원하는 작업을 수행하지 않고 Data Pump가 종료됩니다.

추가적인 새로운 기능에 대한 자세한 내용은 *Oracle Database Utilities 11g Release 2*를 참조하십시오.

Data Pump 기존 모드

Data Pump 엑스포트 및 임포트 유ти리티:

- Data Pump 형식의 파일만 읽기 및 쓰기
- 기존 모드의 `exp` 및 `imp` 유ти리티 명령 허용
- 다음과 같은 기존 모드 파라미터 포함:
 - 새 구문과 동일한 파라미터:
`FILESIZE=integer[B | K | M | G]`
 - 유사한 파라미터:
`QUERY= query_clause`
 - Data Pump 기본값으로 해당 명령이 대체되는 경우 무시되는 파라미터
`BUFFER=integer`
`COMPRESS={y|n}`
`DIRECT={y|n}`
 - 이전 구문 및 새 구문이 혼합된 경우 오류 발생

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Data Pump 기존 모드

Data Pump 유ти리티는 Data Pump 형식의 파일에서만 작동합니다. `exp` 유ти리티로 생성된 파일은 `imp` 유ти리티로 읽어야 합니다. Data Pump 기존 모드에서는 기존 스크립트를 Data Pump 유ти리티에서도 계속 사용할 수 있습니다. 하지만 데이터베이스의 새 기능을 사용하려면 새 Data Pump 구문을 사용해야 합니다.

유ти리티에서 고유한 `exp` 또는 `imp` 파라미터가 발견되면 기존 모드로 전환됩니다.

- 동일한 파라미터는 변경되지 않고 사용됩니다. 예제: `FILESIZE=integer [B | K | M | G]` 파라미터는 덤프 파일의 최대 크기를 지정합니다.
- `QUERY=query_clause` 파라미터는 어떠한 변환도 일으키지 않지만 `expdp` 유ти리티는 지원 중단된 엑스포트 유ти리티보다 덜 제한적인 방식으로 `query`를 처리합니다. 따라서 `query` 결과가 조금 다를 수 있습니다.
- 일부 파라미터는 새 기본값으로 대체되었기 때문에 무시됩니다. 예제:
 - `BUFFER=integer` 파라미터는 `expdp` 유ти리티에 Conventional Path 모드가 없기 때문에 무시됩니다.
 - `COMPRESS={y|n}` 파라미터는 `expdp` 유ти리티에 상응하는 파라미터가 없기 때문에 무시됩니다.
 - `DIRECT={y|n}` 파라미터는 `expdp` 유ти리티가 요청된 엑스포트에 Direct Path 또는 External Table 모드를 사용할지 여부를 결정하기 때문에 무시됩니다.
- `exp/imp` 및 Data Pump 파라미터가 혼합된 경우 작업이 실패합니다.

Data Pump 기존 모드

- **기존 모드 파라미터:**

- Data Pump 파라미터에 매핑됩니다(가능한 경우).

```
consistent={y|n} -> FLASHBACK_TIME  
GRANTS=n -> EXCLUDE=CONSTRAINTS  
INDEXES=n -> EXCLUDE=INDEX  
LOG=filename -> LOGFILE=filename  
FILE=filename -> dumpfile=directory-object:filename
```

- 비슷할 수는 있지만 동일하지 않습니다.

```
FEEDBACK=integer -> STATUS
```

- 새로운 Data Pump와 호환되지 않을 경우 오류가 발생합니다.

```
VOLSIZE=integer
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Pump 기존 모드(계속)

Data Pump 기존 모드를 사용하면 지원이 중단된 스크립트를 현재 버전으로 변환할 수 있습니다. 일부 파라미터는 새 구문으로 매핑할 수 있습니다.

- Data Pump는 현재 시간을 인식하고 CONSISTENT={y|n} 파라미터를 FLASHBACK_TIME 파라미터로 매핑합니다.
- GRANTS=n 파라미터는 EXCLUDE=GRANT로 재매핑됩니다.
- INDEXES=n 파라미터는 EXCLUDE=INDEX로 재매핑됩니다.
- LOG=filename 파라미터는 LOGFILE=filename으로 재매핑됩니다. 로그 파일의 내용(정보 및 오류 메시지)은 expdp 형식으로 표시됩니다.
- FILE=filename 파라미터는 dumpfile=directory-object:filename으로 재매핑됩니다. 하지만 expdp 유ти리티가 기존 디렉토리 객체에 대한 경로를 찾을 수 없는 경우에는 중단됩니다.

파라미터를 매핑할 수 있지만 새로운 기능이 다릅니다. FEEDBACK=integer 파라미터는 STATUS로 재매핑됩니다. 이 매핑은 처리된 행 수가 아닌 엑스포트 작업 상태가 반환되기 때문에 직접 매핑이 아닙니다.

파라미터는 Data Pump와 호환되지 않을 경우 작업을 중단시킬 수 있습니다. exp 유ти리티의 VOLSIZE=integer 파라미터는 테이프 볼륨 크기를 지정합니다. Data Pump에서는 테이프 드라이브를 지원하지 않으며, Oracle Secure Backup에서 테이프를 관리합니다.

Data Pump 유ти리티에 대한 exp 및 imp 파라미터의 전체 매핑에 대해서는 *Oracle Database Utilities 11g Release 2*를 참조하십시오.

파일 위치 관리

- 원본 exp 및 imp 유ти리티: 전체 파일 이름
- 파일 위치에 대한 Data Pump 디렉토리 객체
 - 기본값(이전 버전): DATA_PUMP_DIR 파라미터
 - 새 선택 사항 DATA_PUMP_DIR_schema-name 디렉토리 객체
 - CREATE DIRECTORY 및 GRANT SQL 명령으로 관리
 - 다음 경우에 기본 위치 사용(기준 모드와 독립적):
 - DIRECTORY 파라미터가 없는 명령행
 - EXP_FULL_DATABASE 권한이 없는 유저

ORACLE

Copyright © 2010, Oracle. All rights reserved.

파일 위치 관리

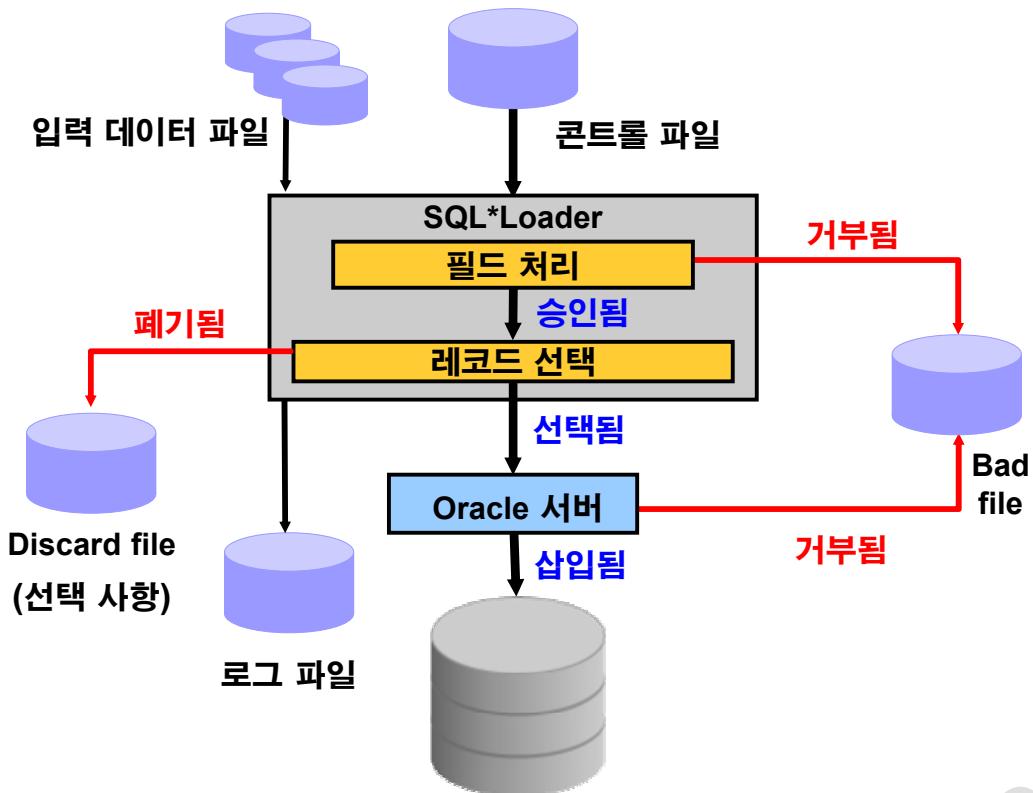
원본 유ти리티는 클라이언트 기반이기 때문에(모든 파일 이름에 전체 이름 사용) 원본 유ти리티 및 Data Pump 유ти리티는 파일 위치에 대한 처리 방법이 다릅니다.

Data Pump 유ти리티는 서버 기반입니다. 이 유ти리티에서는 파일 위치를 지정할 때 디렉토리 객체를 사용해야 합니다. 디렉토리 객체는 스키마에 액세스 할 수 있어야 합니다. 이전 버전에서 기본 위치는 DATA_PUMP_DIR 초기화 파라미터에 의해 설정되었습니다.

이 기능을 사용하기 위해 DBA가 새로운 필수 작업을 수행할 필요는 없습니다. 이 기능은 선택적인 DATA_PUMP_DIR_<schema-name> 디렉토리 객체를 선보입니다. CREATE DIRECTORY 및 GRANT SQL 명령으로 이 디렉토리 객체를 생성하도록 선택한 경우 지정된 스키마에서 Data Pump 익스포트 또는 임포트를 사용할 수 있으므로, CPU, 메모리 사용 및 디스크 사용과 같은 일반적인 서버 리소스에 영향을 줄 수 있습니다.

Data Pump 유ти리티는 명령행에 지정된 항목이 없고 유저에게 EXP_FULL_DATABASE 권한이 없는 경우 이 디렉토리 객체를 사용합니다. 이 유ти리티는 기준 모드와는 독립적입니다.

SQL*Loader: 개요



ORACLE

Copyright © 2010, Oracle. All rights reserved.

SQL*Loader: 개요

SQL*Loader는 데이터를 외부 파일에서 오라클 데이터베이스의 테이블로 로드합니다. SQL*Loader에는 데이터 파일의 데이터 형식을 거의 제한하지 않는 강력한 데이터 구문 분석 엔진이 있습니다.

SQL*Loader는 다음 파일을 사용합니다.

입력 데이터 파일: SQL*Loader는 콘트롤 파일에 지정된 하나 이상의 파일(또는 운영 체제에서 파일에 해당하는 것)에서 데이터를 읽습니다. SQL*Loader의 관점에서 데이터 파일의 데이터는 레코드로 구성됩니다. 특정 데이터 파일은 고정 레코드 형식, 가변 레코드 형식 또는 스트림 레코드 형식이 될 수 있습니다. 레코드 형식은 콘트롤 파일에서 INFILE 파라미터를 사용하여 지정할 수 있습니다. 레코드 형식을 지정하지 않으면 기본값인 스트림 레코드 형식이 사용됩니다.

콘트롤 파일: 콘트롤 파일은 SQL*Loader가 이해할 수 있는 언어로 작성된 텍스트 파일입니다. 콘트롤 파일을 통해 SQL*Loader는 데이터 검색 위치, 데이터 구문 분석 및 해석 방법, 데이터 삽입 위치 등을 알 수 있습니다. 정확하게 정의되어 있지는 않지만 콘트롤 파일은 다음과 같은 세 섹션으로 구분될 수 있습니다.

- 첫번째 섹션에는 다음과 같은 세션 정보가 포함되어 있습니다.
 - 입력 데이터 파일 이름과 같은 Global 옵션 및 건너뛸 레코드
 - 입력 데이터의 위치를 지정하는 INFILE 절
 - 로드할 데이터

SQL*Loader: 개요(계속)

- 두번째 세션은 하나 이상의 INTO TABLE 블록으로 구성됩니다. 이러한 각 블록은 데이터가 로드될 테이블에 대한 정보(예: 테이블 이름, 테이블 열)를 포함합니다.
- 세번째 세션은 선택 사항이며 있을 경우 입력 데이터를 포함합니다.

로그 파일: SQL*Loader가 실행되면 로그 파일을 생성합니다. SQL*Loader가 로그 파일을 생성할 수 없으면 실행이 종료됩니다. 로그 파일은 로드 중 발생한 오류에 대한 설명을 포함하여 로드에 대한 상세한 요약을 포함합니다.

Bad file: SQL*Loader 또는 오라클 데이터베이스에 의해 거부된 레코드를 포함합니다. 입력 형식이 부적합하면 SQL*Loader에서 데이터 파일 레코드를 거부합니다. 데이터 파일 레코드가 SQL*Loader에서 처리하도록 승인되면 테이블에 행으로 삽입하기 위해 오라클 데이터베이스로 전송됩니다. 오라클 데이터베이스에서 행이 적합하다고 판단되면 행이 테이블에 삽입됩니다. 행이 부적합하다고 판단되면 해당 레코드가 거부되고 SQL*Loader는 거부된 레코드를 bad file에 넣습니다.

Discard file: 이 파일은 필요한 경우에만 생성되며 Discard file을 활성화해야 한다고 지정한 경우에만 생성됩니다. Discard file에는 콘트롤 파일에 지정된 레코드 선택 기준과 일치하지 않기 때문에 로드에서 필터링되어 걸러진 레코드가 포함됩니다.

SQL*Loader에 대한 자세한 사항은 *Oracle Database Utilities* 설명서를 참조하십시오.

SQL*Loader로 데이터 로드

The screenshot shows the 'Load Data: Generate Or Use Existing Control File' dialog. It has two options: 'Automatically Generate Control File' (unchecked) and 'Use Existing Control File' (checked). Below this is the 'Host Credentials' section with 'Username' set to 'oracle' and 'Password' set to '*****'. A red arrow points from the 'Control File' step in the navigation bar below to the 'Use Existing Control File' option in this dialog.

Load Data: Generate Or Use Existing Control File

Database orcl.oracle.com

Automatically Generate Control File
A control file will be generated after you define the structure of the data file.

Use Existing Control File
Allows you to use an existing control file that defines the structure of the data file.

Host Credentials

* Username oracle

* Password *****

Save as Preferred Credential

Control File Data File Load Method Options Schedule Review

Load Data: Control File

Database orcl.oracle.com

Cancel Finish Step 1 of 6 Next

A control file is used to describe what will be loaded and how. Specify the full path and name of the control file on the database server machine.

/u01/app/oracle/oradata/orcl2/LOAD.CTL

ORACLE

Copyright © 2010, Oracle. All rights reserved.

SQL*Loader로 데이터 로드

"Load Data from User Files Wizard"를 사용하여 플랫 파일의 데이터를 오라클 데이터베이스로 로드할 수 있습니다.

마법사를 표시하려면 Enterprise Manager Data Movement > Move Row Data > Load Data from User Files를 선택합니다.

SQL*Loader 콘트롤 파일

SQL*Loader 콘트롤 파일은 SQL*Loader에 다음 정보를 알려줍니다.

- 로드할 데이터의 위치
- 데이터 형식
- 구성 세부 사항:
 - 메모리 관리
 - 레코드 거부
 - 중단된 로드 처리 세부 사항
- 데이터 조작 세부 사항



ORACLE

Copyright © 2010, Oracle. All rights reserved.

SQL*Loader 콘트롤 파일

SQL*Loader 콘트롤 파일은 DDL(데이터 정의어) 지침을 포함하는 텍스트 파일입니다.
DDL은 SQL*Loader 세션의 다음 측면을 제어하는 데 사용됩니다.

- SQL*Loader가 로드할 데이터를 찾는 위치
- SQL*Loader에 필요한 데이터 형식
- SQL*Loader가 데이터를 로드할 때 메모리 관리, 선택 및 거부 기준, 중단된 로드 처리 등을 구성하는 방법
- SQL*Loader가 로드 중인 데이터를 조작하는 방법

SQL*Loader 콘트롤 파일(계속)

```
1 -- This is a sample control file
2 LOAD DATA
3 INFILE 'SAMPLE.DAT'
4 BADFILE 'sample.bad'
5 DISCARDFILE 'sample.dsc'
6 APPEND
7 INTO TABLE emp
8 WHEN (57) = '.'
9 TRAILING NULLCOLS
10 (hiredate SYSDATE,
     deptno POSITION(1:2) INTEGER EXTERNAL(3)
     NULLIF deptno=BLANKS,
           job POSITION(7:14) CHAR TERMINATED BY WHITESPACE
     NULLIF job=BLANKS "UPPER(:job)",
           mgr POSITION(28:31) INTEGER EXTERNAL
           TERMINATED BY WHITESPACE, NULLIF mgr=BLANKS,
           ename POSITION(34:41) CHAR
           TERMINATED BY WHITESPACE "UPPER(:ename)",
           empno POSITION(45) INTEGER EXTERNAL
           TERMINATED BY WHITESPACE,
           sal POSITION(51) CHAR TERMINATED BY WHITESPACE
           "TO_NUMBER(:sal,'$99,999.99')",
           comm INTEGER EXTERNAL ENCLOSED BY '(' AND '%'
           ":comm * 100"
      )
```

이 샘플 콘트롤 파일을 행 번호별로 설명하면 다음과 같습니다.

1. 주석은 파일의 명령 셝션 어디에나 나타날 수 있지만 데이터 내에 나타나면 안됩니다.
주석 앞에는 두 개의 하이픈(-)을 넣습니다. 두 개의 하이픈 오른쪽에 있는 모든 텍스트는 해당 행의 끝까지 무시됩니다.
2. LOAD DATA 문은 여기서부터 새 데이터 로드가 시작됨을 SQL*Loader에 알립니다.
진행 중에 중단된 로드를 계속할 경우에는 CONTINUE LOAD DATA 문을 사용합니다.
3. INFILE 키워드는 로드하려는 데이터를 포함하는 데이터 파일의 이름을 지정합니다.
4. BADFILE 키워드는 거부된 레코드를 배치할 파일 이름을 지정합니다.
5. DISCARDFILE 키워드는 폐기된 레코드를 배치할 파일 이름을 지정합니다.
6. APPEND 키워드는 비어 있지 않은 테이블로 데이터를 로드할 때 사용할 수 있는 옵션
중 하나입니다. 데이터를 비어 있는 테이블로 로드하려면 INSERT 키워드를 사용합니다.
7. INTO TABLE 키워드는 테이블, 필드 및 데이터 유형을 식별할 수 있습니다. 이 키워드는
데이터 파일에 있는 레코드와 데이터베이스에 있는 테이블 간의 관계를 정의합니다.
8. WHEN 절은 SQL*Loader가 데이터를 로드하기 전에 각 레코드가 충족해야 하는 필드 조건을
하나 이상 지정합니다. 이 예제에서는 SQL*Loader가 57번째 문자가 소수점인 경우에만
레코드를 로드합니다. 이 소수점은 필드에서 달러와 센트를 구분하며 SAL에 값이 없을
경우 레코드를 거부하도록 합니다.
9. TRAILING NULLCOLS 절은 SQL*Loader가 레코드에 없는 상대적으로 배치된 열을 NULL
열로 처리하게 합니다.
10. 콘트롤 파일의 나머지 부분에는 로드 중인 테이블의 열 형식에 대한 정보를 제공하는 필드
리스트가 있습니다.

로드 방식



Conventional Load	Direct Path Load
COMMIT 사용	Data Save 사용(속도가 더 빠름)
항상 리두 항목을 생성합니다.	특정 조건에서만 리두를 생성합니다.
모든 제약 조건을 적용합니다.	PRIMARY KEY, UNIQUE 및 NOT NULL만 강제 시행
INSERT 트리거를 실행합니다.	INSERT 트리거를 실행하지 않습니다.
클러스터화된 테이블로 로드할 수 있습니다.	클러스터로 로드하지 않습니다.
로드 작업 동안 다른 유저가 테이블을 수정할 수 있습니다.	로드 작업 동안 다른 유저가 테이블을 변경할 수 없습니다.
각 삽입 시 인덱스 항목 유지	로드 종료 시 새 인덱스 항목 병합

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Direct Path Load와 Conventional Path Load 비교

데이터 저장 방식

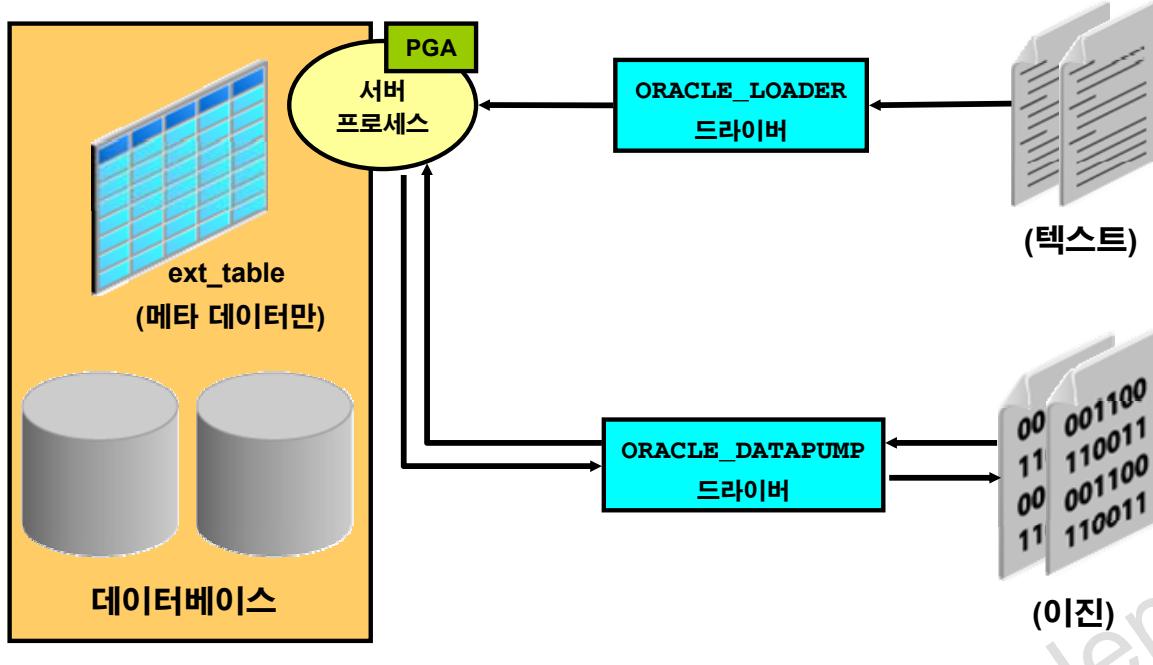
Conventional Path Load는 SQL INSERT 문을 실행하여 오라클 데이터베이스에 있는 테이블을 채웁니다. Direct Path Load는 오라클 데이터 블록의 형식을 지정하고 데이터 블록을 데이터베이스 파일에 직접 기록하여 오라클 데이터베이스의 오버헤드를 상당 부분 제거합니다. Direct Load는 다른 유저와 데이터베이스 리소스를 경합하지 않으므로 일반적으로 종료 시 데이터를 디스크 속도로 로드할 수 있습니다. Conventional path load는 SQL 프로세스 및 데이터베이스 COMMIT 작업을 사용하여 데이터를 저장합니다. 일련의 레코드 삽입 다음에는 COMMIT 작업이 따라야 합니다. 각 데이터 로드에는 여러 트랜잭션이 관련될 수 있습니다.

Direct path load는 data saves를 사용하여 데이터 블록을 Oracle 데이터 파일에 기록합니다. Direct path load가 Conventional path load보다 속도가 빠른 이유는 이 때문입니다. COMMIT과 구별되는 data save의 특징은 다음과 같습니다.

- Data Save 중 전체 데이터베이스 블록만 데이터베이스에 기록됩니다.
- 이 블록은 테이블의 HWM(high-water mark) 다음에 기록됩니다.
- Data Save 이후에는 HWM이 이동됩니다.
- data save 후에 내부 리소스가 해제되지 않습니다.
- Data Save가 트랜잭션을 종료하지 않습니다.
- 각 data save마다 인덱스가 갱신되지 않습니다.

External Table

External Table은 오라클 데이터베이스 외부의 운영 체제 파일로 저장되는 읽기 전용 테이블입니다.



Copyright © 2010, Oracle. All rights reserved.

External Table

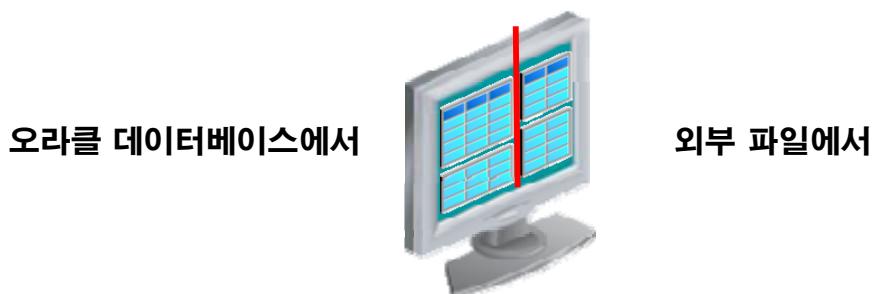
External Table은 데이터베이스 테이블과 같은 방식으로 외부 소스의 데이터를 액세스합니다. 데이터베이스에 연결하고 DDL을 사용하여 External Table에 대한 메타 데이터를 생성할 수 있습니다. External Table에 대한 DDL은 두 부분으로 구성됩니다. 한 부분에서는 오라클 데이터베이스 열 유형을 기술하고, 다른 부분에서는 오라클 데이터베이스 테이터 열에 대한 외부 데이터의 매핑을 기술합니다.

External Table은 데이터베이스에 저장된 데이터를 나타내지 않으며 외부 소스에 데이터를 저장하는 방법을 나타내지도 않습니다. 대신 External Table 층에서 서버에 데이터를 어떻게 표시해야 하는지를 나타냅니다. External Table 정의에 맞게 외부 파일의 데이터에 필요한 변형을 수행하는 작업은 액세스 드라이버와 External Table 층에서 담당합니다. External Table은 읽기 전용이므로 DML 작업을 수행할 수 없으며, 인덱스를 생성할 수도 없습니다.

External Table에는 두 가지 액세스 드라이버가 있습니다. ORACLE_LOADER 액세스 드라이버는 External Table에서 테이블 데이터를 읽는 데에만 사용할 수 있으며, 이를 데이터베이스에 로드합니다. 여기에서는 텍스트 파일을 데이터 소스로 사용합니다. ORACLE_DATAPUMP 액세스 드라이버는 외부 파일의 테이블 데이터를 데이터베이스에 로드하고 데이터베이스의 데이터를 외부 파일로 로드 취소할 수도 있습니다. 여기에서는 이진 파일을 외부 파일로 사용합니다. 이진 파일은 impdp 및 expdp 유ти리티에서 사용되는 파일과 형식이 동일하며, 서로 바꿔서 사용할 수 있습니다.

External Table 이점

- 외부 파일에서 직접 데이터를 사용하거나 다른 데이터베이스로 데이터를 로드할 수 있습니다.
- 외부 데이터는 먼저 로드할 필요 없이 데이터베이스에 상주하는 테이블과 병렬로 query 및 조인할 수 있습니다.
- 복합 query의 결과는 외부 파일로 로드 취소할 수 있습니다.
- 로드를 위해 다른 소스에서 생성된 파일을 결합할 수 있습니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

External Table 이점

External Table을 위해 생성된 데이터 파일은 동일한 데이터베이스 또는 다른 데이터베이스에 있는 다른 External Table을 위해 데이터 파일로 이동되거나 사용될 수 있습니다. 외부 데이터는 먼저 데이터를 로드할 필요 없이 데이터베이스에 상주하는 테이블과 병렬로 query 및 조인할 수 있습니다. SELECT 명령을 사용하여 응용 프로그램이 External Table에 직접 액세스하도록 할 수도 있고, 데이터가 먼저 Target Database로 로드되도록 할 수도 있습니다.

복합 query의 결과는 ORACLE_DATAPUMP 액세스 드라이버를 사용하여 외부 파일로 로드 취소할 수 있습니다.

서로 다른 External Table에 의해 채워진 데이터 파일은 다른 External Table의 LOCATION 절에 모두 지정될 수 있습니다. 따라서 여러 소스의 데이터를 쉽게 집계할 수 있습니다. 단, 모든 External Table의 메타 데이터가 정확하게 동일해야 한다는 제한이 있습니다.

ORACLE_LOADER로 External Table 정의

```
CREATE TABLE extab_employees
  (employee_id          NUMBER(4),
   first_name           VARCHAR2(20),
   last_name            VARCHAR2(25),
   hire_date             DATE)
ORGANIZATION EXTERNAL
  ( TYPE ORACLE_LOADER DEFAULT DIRECTORY extab_dat_dir
    ACCESS PARAMETERS
      ( records delimited by newline
        badfile extab_bad_dir:'empxt%a_%p.bad'
        logfile extab_log_dir:'empxt%a_%p.log'
        fields terminated by ','
        missing field values are null
      ( employee_id, first_name, last_name,
        hire_date char date_format date mask "dd-mon-yyyy"))
    LOCATION ('empxt1.dat', 'empxt2.dat') )
PARALLEL REJECT LIMIT UNLIMITED;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

External Table을 ORACLE_LOADER

로 정의 External Table의 메타 데이터는 데이터베이스에서 SQL 언어를 사용하여 생성됩니다. ORACLE_LOADER 액세스 드라이버는 SQL*Loader 구문을 사용하여 External Table을 정의합니다. 이 명령은 외부 텍스트 파일을 생성하지 않습니다. 슬라이드의 예제에서는 생성된 후 유저에게 액세스 권한이 부여된 기존 OS 디렉토리로 매핑되는 세 개의 디렉토리 객체(extab_dat_dir, extab_bad_dir 및 extab_log_dir)를 보여줍니다.

extab_employees 테이블이 액세스되면 SQL*Loader 기능을 사용하여 테이블을 로드하고 해당 Instance에서 로그 파일 및 bad file이 생성됩니다.

최적의 사용법(Best-Practice Tip): 로드할 데이터가 많은 경우 로드 작업에 대해 PARALLEL을 활성화하십시오.

```
ALTER SESSION ENABLE PARALLEL DML;
```

ORACLE_DATAPUMP로 External Table 채우기

```
CREATE TABLE ext_emp_query_results
  (first_name, last_name, department_name)
ORGANIZATION EXTERNAL
(
  TYPE ORACLE_DATAPUMP
  DEFAULT DIRECTORY ext_dir
  LOCATION ('emp1.exp', 'emp2.exp', 'emp3.exp')
)
PARALLEL
AS
SELECT e.first_name, e.last_name, d.department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id AND
      d.department_name in
        ('Marketing', 'Purchasing');
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

ORACLE_DATAPUMP로 External Table 채우기

이 예제는 새로운 External Table 채우기 작업을 통해 EMPLOYEES 및 DEPARTMENTS 테이블의 조인 결과로 나온 레코드 집합을 선택적으로 엑스포트하는 방법을 보여줍니다.

External Table의 크기가 매우 클 수 있기 때문에 데이터를 External Table에 로드 취소하는 데 Parallel Populate 작업을 사용할 수 있습니다. External Table의 Parallel Query와 달리 Parallel Populate 작업의 병렬도는 액세스 드라이버에 의해 기록되는 동시 파일 수에 의해 제약을 받습니다. 두 대 이상의 병렬 실행 서버가 특정 시점에서 한 파일에 기록할 수 없습니다.

각 입출력(I/O) 서버 프로세스는 자체의 파일을 필요로 하므로 LOCATION 절에 있는 파일의 수는 지정된 병렬도와 일치해야 합니다. 지정된 임의의 추가 파일은 무시됩니다. 지정된 병렬도에 맞게 충분한 파일이 없을 경우 LOCATION 절에 있는 파일 수와 일치하도록 병렬도가 낮아집니다.

External Table은 채워진 후 읽기 전용이 됩니다. SELECT 명령은 매우 복잡하게 구성하여 External Table에 특정 정보를 채울 수 있습니다. 그런 다음 이진 Data Pump 파일과 파일 구조가 동일한 External Table을 다른 시스템으로 이전하고 impdp 유ти리티로 임포트되거나 External Table로 읽힐 수 있습니다.

주: ORACLE_DATAPUMP 액세스 드라이버 파라미터에 대한 자세한 내용은 *Oracle Database Utilities 11g Release 2 (11.2)* 설명서를 참조하십시오.

External Table 사용

- External Table 조회:

```
SQL> SELECT * FROM extab_employees;
```

- External Table 조회 및 External Table과 Internal Table 조인

```
SQL> SELECT e.employee_id, e.first_name, e.last_name,
d.department_name FROM departments d, extab_employees e
WHERE d.department_id = e.department_id;
```

- External Table에서 Internal Table로 데이터 추가

```
SQL> INSERT /*+ APPEND */ INTO hr.employees SELECT * FROM
extab_employees;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

External Table 사용

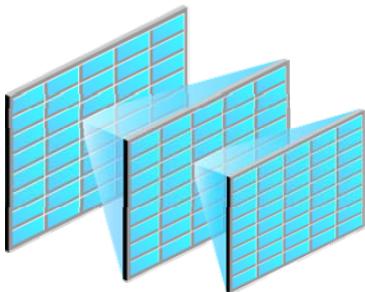
External Table은 Internal 데이터베이스 테이블과 같은 방식으로 query됩니다. 첫번째 예제에서는 External Table EXTAB_EMPLOYEES를 query하고 결과를 표시만 하는 것을 보여 줍니다. query 결과는 데이터베이스에 저장되지 않습니다.

두번째 예제에서는 Internal Table DEPARTMENTS를 External Table EXTAB_EMPLOYEES와 조인하고 결과를 표시만 하는 것을 보여 줍니다. 슬라이드의 세번째 예제에서는 query로 Internal Table 데이터를 직접 추가하고 External Table에서 데이터를 로드하는 것을 보여 줍니다.

데이터 딕셔너리

External Table에 대한 정보 보기:

- [DBA | ALL | USER]_EXTERNAL_TABLES
- [DBA | ALL | USER]_EXTERNAL_LOCATIONS
- [DBA | ALL | USER]_TABLES
- [DBA | ALL | USER]_TAB_COLUMNS
- [DBA | ALL]_DIRECTORIES



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터 딕셔너리

슬라이드에 나와 있는 데이터 딕셔너리 뷰에는 다음과 같은 테이블 정보가 나열됩니다.

- [DBA | ALL | USER]_EXTERNAL_TABLES: 데이터베이스에 있는 External Table의 특정 속성
- [DBA | ALL | USER]_EXTERNAL_LOCATIONS: External Table의 데이터 소스
- [DBA | ALL | USER]_TABLES: 데이터베이스에 있는 관계형 테이블의 설명
- [DBA | ALL | USER]_TAB_COLUMNS: 데이터베이스에 있는 테이블, 뷰 및 클러스터 열의 설명
- [DBA | ALL]_DIRECTORIES: 데이터베이스에 있는 디렉토리 객체의 설명

퀴즈

다른 데이터베이스 객체와 마찬가지로 디렉토리 객체는 생성 중 다른 스키마가 지정되지 않는 한 이를 생성하는 유저에 의해 소유됩니다.

1. 참
2. 거짓

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 2

퀴즈

인덱스는 External Table에 생성할 수 있습니다.

1. 참
2. 거짓



Copyright © 2010, Oracle. All rights reserved.

해답: 2

Oracle Internal & Oracle Academy
Use Only

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 데이터 이동 방법 설명
- 디렉토리 객체 생성 및 사용
- SQL*Loader를 사용하여 비오라클 데이터베이스 또는 유저 파일에서 데이터 로드
- External Table을 사용하여 플랫폼 독립적인 파일을 통해 데이터 이동
- Oracle Data Pump의 일반적 구조 설명
- Data Pump 엑스포트/임포트를 사용하여 오라클 데이터베이스 간에 데이터 이동

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

연습 17 개요: 데이터 이동

이 연습에서는 다음 내용을 다룹니다.

- Data Pump Export Wizard를 사용하여 엑스포트할 데이터베이스 객체 선택
- Data Pump 엑스포트 작업 모니터
- Data Pump Import Wizard를 사용하여 데이터베이스에 테이블 임포트
- Load Data Wizard를 사용하여 데이터베이스로 데이터 로드
- 명령행을 사용하여 데이터 로드

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

18

지원 활용

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

목표

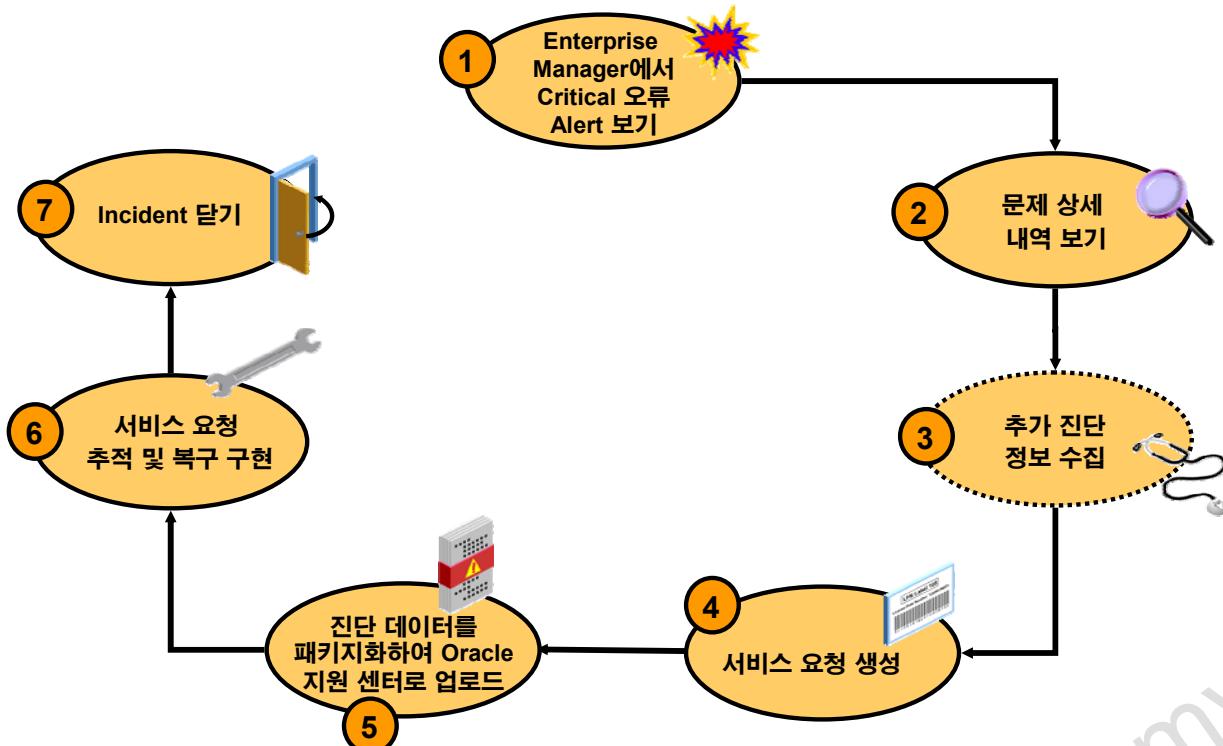
이 단원을 마치면 다음을 수행할 수 있습니다.

- Enterprise Manager Support Workbench 사용
- My Oracle Support 활용
- My Oracle Support 검색
- SR(서비스 요청) 로깅
- 패치 관리
 - 패치 적용
 - 패치 스테이지

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Support Workbench 사용



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Support Workbench 사용

Enterprise Manager Support Workbench를 사용하면 다음과 같은 일반 단계를 수행하여 문제를 조사, 보고 및 해결(일부 경우)할 수 있습니다.

1. Enterprise Manager의 Database Home 페이지에서 Critical 오류 Alert를 검토합니다. Alert를 선택하여 세부 정보를 확인합니다.
2. 문제 세부 정보를 점검하고 문제에 대해 기록된 모든 Incident list를 확인합니다. 자동으로 실행된 모든 상태 검사의 결과를 표시합니다.
3. 선택적으로 추가 상태 검사를 실행하고 SQL Test Case Builder를 호출합니다. 이 도구는 SQL 문제와 관련된 모든 필수 데이터를 수집하고 Oracle 지원 센터에서 문제를 재생산할 수 있는 방식으로 정보를 패키지화합니다.
4. My Oracle Support를 통해 서비스 요청을 생성하고 원하는 경우 문제 정보를 사용하여 서비스 요청 번호를 기록합니다.
5. Incident Packaging Service를 호출합니다. 이 도구는 문제에 대해 수집된 모든 진단 데이터를 패키지화하고 선택적으로 해당 데이터를 Oracle 지원 센터에 업로드합니다. 업로드 전에 데이터를 편집하여 중요한 정보를 제거할 수 있습니다.
6. Support Workbench에서 서비스 요청에 대한 작업 로그를 유지 관리할 수 있습니다. Oracle Advisor를 실행하여 SQL Failure 또는 손상된 데이터를 복구합니다.
7. 문제에 대한 하나, 일부 또는 모든 Incident의 상태가 닫히도록 설정합니다.

Enterprise Manager에서 Critical 오류 Alert 보기

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. On the left, there's a sidebar with 'General' and 'Diagnostic Summary' sections. Under 'General', it shows 'Status Up', 'Up Since Apr 2, 2007 12:28:32', 'Instance Name db', 'Version 11.1.0.4.0', 'Host stac017.us.oracle.com', and 'Listener LISTENER stac017.us.oracle.com'. There's a link 'View All Properties'. Under 'Diagnostic Summary', it says 'No ADDM run available' and 'No ARA errors'. A red box highlights the 'Active Incidents' link. Below the sidebar, there's a 'Support Workbench' section with tabs for 'Problems (5)', 'Checker Findings (36)', and 'Packages (4)'. The 'Problems (5)' tab is selected. It shows statistics: 'New Problems in Last 24 Hours 0', 'All Active Problems 1', 'All Problems 5'; 'New Incidents in Last 24 Hours 2', 'All Active Incidents 2', 'All Incidents 43'. It also has a 'View' dropdown set to 'Last 24 Hours' and a search bar. A red box highlights the 'Incidents (36)' link. Below this, there's a table with columns 'ID', 'Description', 'Number Of Incidents', 'Last Incident', and 'Last Comment'. One row is shown: ID 1, Description ORA-1578 [1][63671], Number Of Incidents 36, Last Incident April 9, 2007 10:00:24 PM PDT, Last Comment 'Created package : Id = 5 Name = Pkg_database_ORA_1578_040607125612 Yes'. A red box highlights the 'Hide' link next to the ID. The table also has 'Active', 'Packed', and 'SR#' columns. At the bottom of the Support Workbench, there are 'Related Links' for 'Advisor Central', 'Create User-Reported Problem', 'Alert Log Contents', 'Incident Packaging Configuration', and 'Alert Log Errors'. A red arrow points from the 'Active Incidents' link in the sidebar to the 'Incidents (36)' link in the Support Workbench.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager에서 Critical 오류 Alert 보기

문제(Critical 오류) 조사 프로세스에서는 먼저 Database Home 페이지에서 Critical 오류 Alert를 검토합니다. Critical 오류 Alert를 보려면 Enterprise Manager의 Database Home 페이지에 액세스합니다. Home 페이지에서 Diagnostic Summary 섹션의 Active Incidents 링크를 누를 수 있습니다(Incident가 있는 경우). 또한 Alerts 섹션에서 Incidents로 플래그가 지정된 Critical Alert를 확인할 수 있습니다.

Active Incidents 링크를 누르면 모든 문제 및 그에 해당하는 Incident에 대한 세부 정보를 검색할 수 있는 Support Workbench 페이지에 액세스하게 됩니다. 이 페이지에서는 모든 Health Monitor 검사 프로그램의 실행 결과와 생성된 패키지를 검색할 수도 있습니다.

주: 이 섹션에서 설명하는 작업은 모두 Enterprise Manager에서 수행됩니다. 이러한 모든 작업은 ADRCI 명령행 유ти리티를 사용하여 수행할 수도 있습니다. ADRCI 유ти리티에 대한 자세한 사항은 *Oracle Database Utilities* 설명서를 참조하십시오.

문제 세부 정보 보기

The screenshot shows the Oracle Support Workbench interface. On the left, there's a sidebar with 'Support Workbench' and sections for 'Problems (5)', 'Incidents (16)', 'Performance', and 'Related Links'. The main area is titled 'Problem Details: ORA 1578'. It displays summary information like SR# (1234), Bug# (--), Active (Yes), and Packaged (Yes). It also shows the number of incidents (36) and the first incident timestamp (April 2, 2007 12:27:37 PM PDT). Below this is the 'Last Incident' section with a timestamp of April 9, 2007 10:00:24 PM PDT, an incident source of 'System Generated', and impact level 0. It also shows checker runs and findings. To the right, under 'Investigate and Resolve', there are sections for 'Assess Damage' (Checker Findings, Run Checkers, Database Instance Health), 'Diagnose' (Alert Log, Related Problems Across Topology, Diagnostic Dumps for Last Incident, Go to Metalink and Research), and 'Resolve' (SQL Repair Advisor, Data Recovery Advisor). At the bottom, there's an 'Activity Log' table showing comments, actions, descriptions, and timestamps for various database operations. A red box highlights the 'Activity Log' tab and the incident ID 13953 in the table.

문제 세부 정보 보기

Support Workbench 페이지의 Problems 하위 페이지에서 조사할 문제의 ID를 누릅니다. 그러면 해당하는 Problem Details 페이지가 나타납니다.

이 페이지에는 문제와 관련된 모든 Incident가 표시됩니다. 문제를 My Oracle Support 서비스 요청 및 버그 번호와 연관시킬 수 있습니다. 이 페이지의 "Investigate and Resolve" 섹션에는 해당 문제에 대해 수행할 수 있는 작업에 대한 직접 링크가 포함된 Self Service 하위 페이지가 있습니다. 그리고 같은 섹션에 있는 Oracle Support 하위 페이지에는 My Oracle Support에 대한 직접 링크가 있습니다.

Activity Log 하위 페이지에는 현재까지 문제에 대해 수행된 시스템 생성 작업이 표시됩니다. 이 하위 페이지를 통해 문제를 조사하는 중에 의견을 추가할 수 있습니다.

Incidents 하위 페이지에서는 관련 Incident ID를 눌러 해당하는 Incident Details 페이지에 액세스할 수 있습니다.

Incident 세부 정보 보기: 덤프 파일

Contents: b_ora_15201_i3953.trc

File /ade/aime_emdbsa_b/oracle/log/diag/rdbms/b/b/incident/incdir_3953/b_ora_15201_i3953.trc
Modified April 2, 2007 12:27:40 PM PDT
Size 3.28 MB

Incident Details: 3953

Summary

Problem Key	ORA-1578
	[1]
	[63671]
Status	Ready
Active	No
Timestamp	April 2, 2007 12:27:33
Impact	Unknown
ECID	Unknown
Data Dumped	No
Source	System Generated
Correlation Keys	SID = 129.232, Proc# = 16777216, Client ProcId = oracle@stacg17 (TNS V1-V3).15201_3083220672
Purge Date	April 22, 2007 5:30:24 AM PDT (Purging Enabled) Disable Purging

Trace Map
A Trace Map provides a table of contents for a dump file.
 TIP Select a section to see its detailed trace records below.

Details

[Expand All](#) | [Collapse All](#)

[/ade/aime_emdbsa_b/oracle/log/diag/rdbms/b/b/incident/incdir_3953/b_ora_15201_i3953.trc](#)

[Error Stack:](#)

incident_dump:===== Dump for incident 3953 (ORA 1578) ======

ORA-01578: ORACLE data block corrupted (file # 1, block # 63671)
ORA-01110: data file 1: '/ade/aime_emdbsa_b/oracle/dbs/t_db1.f'

Dump Files [Checker Findings](#) [Additional Diagnostics](#)

File Name	Size (MB)	Timestamp	Path	View Contents
b_ora_15201_i3953.trc	3.28	April 2, 2007 12:27:40 PM PDT	/ade/aime_emdbsa_b/oracle/log/diag/rdbms/b/b/incident/incdir_3953	
b_m000_15218_i3953_87.trc	0.15	April 2, 2007 12:27:42 PM PDT	/ade/aime_emdbsa_b/oracle/log/diag/rdbms/b/b/incident/incdir_3953	

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Incident 세부 정보 보기: 덤프 파일

Incident Details 페이지에 액세스하면 Dump Files 하위 페이지에 모든 해당 덤프 파일이 나열됩니다. 여기서 특정 덤프 파일의 안경 아이콘을 누르면 다양한 섹션에서 파일 컨텐트를 표시할 수 있습니다.

Incident 세부 정보 보기: 검사 프로그램 실행 결과

The screenshot shows the Oracle Database 11g Incident Details page for Incident ID 3953. The 'Checker Findings' tab is selected. A red arrow points to the 'Select Description' table, which lists a critical finding about a corrupt datafile.

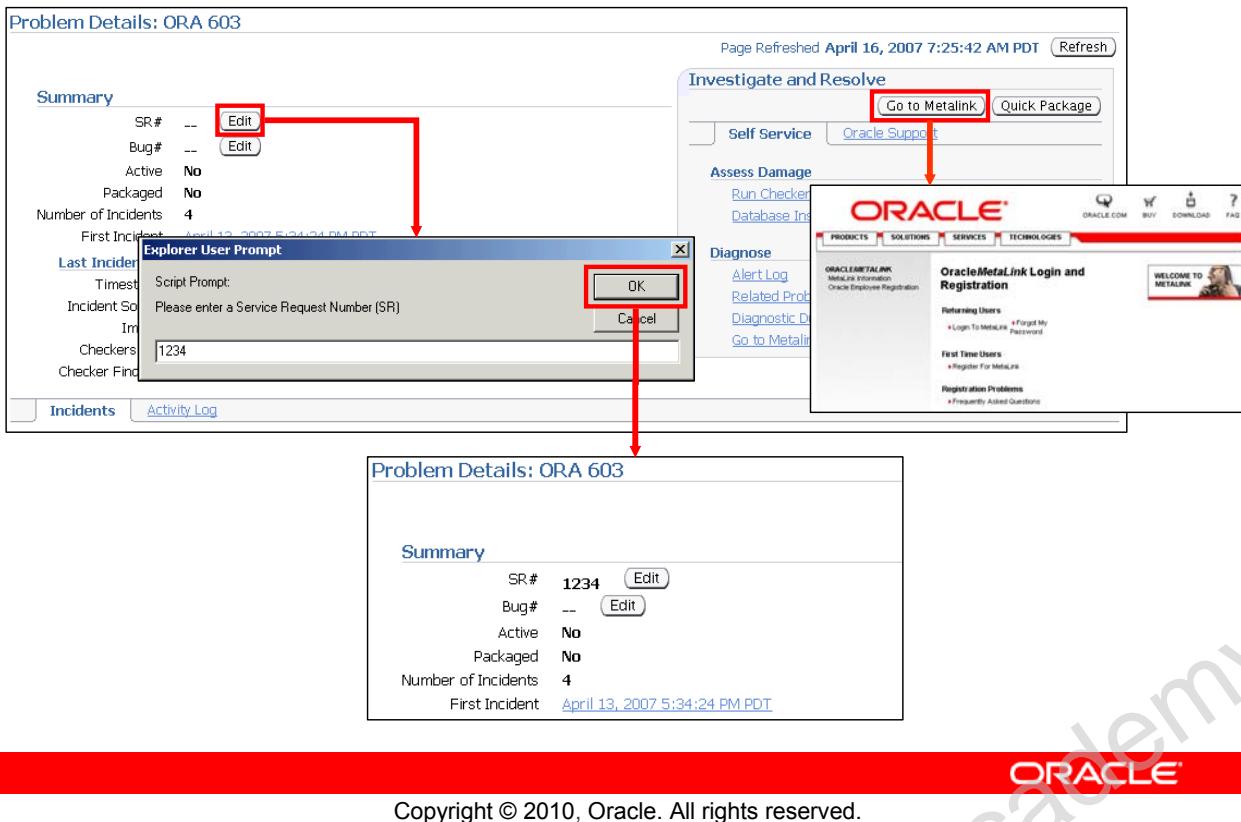
Select Description	Priority	Damage Translation	Incident ID	Status	Time Detected
<input type="checkbox"/> All Findings					
<input type="checkbox"/> Datafile 1: '/ade/aiine_emdbsa_b/oracle/dbs/t_db1.f' contains one or more corrupt blocks	High	Some objects in tablespace SYSTEM might be unavailable	3953	Open	April 2, 2007 12:27:41 PM PDT

ORACLE
Copyright © 2010, Oracle. All rights reserved.

Incident 세부 정보 보기: 검사 프로그램 실행 결과

Incident Details 페이지에서 Checker Findings를 누르면 Checker Findings 하위 페이지가 표시됩니다. 이 페이지에는 Critical 오류가 감지되었을 때 자동으로 실행된 모든 상태 검사의 결과가 표시됩니다. 일반적으로는 하나 이상의 결과를 선택하고 Advisor를 호출하여 문제를 해결할 수 있습니다.

서비스 요청 생성



서비스 요청 생성

문제에 대한 진단 정보를 패키지화하여 Oracle 지원 센터에 업로드하려면 먼저 SR(서비스 요청)을 생성해야 합니다. 서비스 요청을 생성하려면 먼저 My Oracle Support에 액세스합니다.

My Oracle Support는 Problem Details 페이지의 "Investigate and Resolve" 섹션에서 "Go to My Oracle Support" 버튼을 누르면 직접 액세스할 수 있습니다. My Oracle Support가 열리면 일반적인 방식으로 로그인하여 서비스 요청을 생성합니다.

생성이 완료되면 문제에 대해 해당 서비스 요청을 입력할 수 있습니다. 이는 선택적인 작업이며 참조용으로만 사용됩니다.

Summary 섹션에서 SR# 레이블 옆에 있는 Edit 버튼을 누릅니다. Window가 열리면 SR 번호를 입력하고 OK를 누릅니다.

진단 데이터를 패키지화하여 Oracle 지원 센터에 업로드

The screenshot shows the Oracle Support Workbench interface. At the top, there's a 'Problem Details: ORA 603' section with fields for SR# (1234), Bug# (--), and Active (No). To the right is the 'Investigate and Resolve' section with tabs for 'Self Service' and 'Oracle Support'. A red box highlights the 'Quick Package' button. Below this is the 'Quick Packaging: Create New Package' step, labeled 'Step 1 of 4'. It contains fields for Target database (ORA 603), Package Name (Pkg_database_ORA_603), and a note about generating an upload file. A red box highlights the 'Send to Oracle Support' checkbox. The 'Next' button is also highlighted with a red box. To the right is the 'Quick Packaging: View Contents' step, showing a table of incidents to be packaged. The table has columns for ID, Type, Problem ID, and Description. The first few rows show various Oracle errors like ORA-603 and ORA-1578.

진단 데이터를 패키지화하여 Oracle 지원 센터에 업로드

Support Workbench에서는 Incident 패키지를 생성 및 업로드하기 위한 두 가지 방법(빠른 패키징 방법 및 고급 패키징 방법)을 제공합니다. 슬라이드의 예제에서는 빠른 패키징을 사용하는 방법을 보여줍니다.

빠른 패키징은 최소한의 단계로 구성된 자동 방식입니다. 단일 문제를 선택하고 Incident 패키지 이름 및 설명을 입력한 다음 Incident 패키지 업로드 일정을 지정(즉시 업로드 또는 지정된 날짜 및 시간에 업로드)합니다. Support Workbench는 문제와 관련된 진단 데이터를 자동으로 Incident 패키지에 배치하고, Incident 패키지를 종결하며, ZIP 파일을 생성한 다음 해당 파일을 업로드합니다. 이 방법을 사용하면 Incident 패키지 파일을 추가, 편집 또는 제거할 수 없으며 SQL 테스트 사례 등의 다른 진단 데이터를 추가할 수도 없습니다.

진단 데이터를 패키지로 만들어 Oracle 지원 센터에 업로드하려면 다음과 같이 하십시오.

1. Problem Details 페이지의 Investigate and Resolve 섹션에서 Quick Package를 누릅니다. Quick Packaging 마법사의 Create New Package 페이지가 나타납니다.
2. 패키지 이름 및 설명을 입력합니다.
3. 문제를 식별하기 위한 서비스 요청 번호를 입력합니다.
4. Next를 누른 다음 Quick Packaging 마법사의 나머지 페이지를 진행합니다. Review 페이지에서 Submit을 눌러 패키지를 업로드합니다.

서비스 요청 추적 및 복구 구현

Problem Details: ORA 603

Page Refreshed April 16, 2007 8:39:07 AM PDT [Refresh](#)

Summary	
SR #	1234 Edit
Bug#	-- Edit
Active	No
Packaged	Yes
Number of Incidents	4
First Incident	April 13, 2007 5:34:24 PM PDT
Last Incident	
Timestamp	April 13, 2007 6:40:24 PM PDT
Incident Source	System Generated
Impact	
Checkers Run	0
Checker Findings	0

[Investigate and Resolve](#) [Go to Metalink](#) [Quick Package](#)

[Self Service](#) [Oracle Support](#)

[Collect and Send Diagnostic Data](#)

[Create a Service Request with Metalink](#)
[Record Service Request Number to Problem](#)
[Generate Additional Dumps and Test Cases](#)
[Package the Problem](#)
[View/Send Upload Files](#)

[Track and Close](#)

[Check the Service Request Status with Metalink](#)
[Close the problem](#)

[Incidents](#) [Activity Log](#)

Comment	Add Comment		
User	Action	Description	Timestamp
SYS	Comment	Set SR : 1234	April 16, 2007 8:34:45 AM PDT
SYS	Comment	Set SR : null	April 16, 2007 8:34:30 AM PDT
SYS	Package	Failed to send upload file to Oracle: packageId = 1 file = /ade/airme_emrdbsa_b/oracle/stacg17.us.oracle.com_b/sysman/emd/state/Pkg_database_ORA_603_041607080712_COM_1.zip	April 16, 2007 8:14:12 AM PDT
SYS	Package	Created physical file : packageId = 1 file = /ade/airme_emrdbsa_b/oracle/stacg17.us.oracle.com_b/sysman/emd/state/Pkg_database_ORA_603_041607080712_COM_1.zip	April 16, 2007 8:14:10 AM PDT
SYS	Comment	Created package : Id = 1 Name = Pkg_database_ORA_603_041607080712	April 16, 2007 8:09:30 AM PDT

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

서비스 요청 추적 및 복구 구현

진단 정보를 Oracle 지원 센터로 업로드한 후에는 다양한 작업을 수행하여 서비스 요청을 추적하고 복구를 구현할 수 있습니다. 이러한 작업에는 다음이 포함됩니다.

- 문제 정보에 오라클 버그 번호를 추가합니다. Problem Details 페이지에서 Bug# 레이블 옆에 있는 Edit 버튼을 누릅니다. 버그 번호는 참조용으로만 사용됩니다.
- 문제 작업 로그에 의견을 추가하는 작업:
 - 문제에 대한 Problem Details 페이지에 액세스합니다.
 - Activity Log를 눌러 Activity Log 하위 페이지를 표시합니다.
 - Comment 필드에 의견을 입력하고 Add Comment를 누릅니다. 의견이 작업 로그에 기록됩니다.
- Oracle 지원 센터의 추가 진단 제공 요청에 응답합니다. Oracle 지원 센터 담당자가 추가 진단을 수집하여 업로드하라는 지침을 제공할 수 있습니다.

서비스 요청 추적 및 복구 구현

Incident Details: 3953

Page Refreshed April 10, 2007 8:08:22 AM PDT [Refresh](#)

Summary

Problem Key	ORA-1578
[1] [63671]	
Status	Ready
Active	No
Timestamp	April 2, 2007 12:27:37 PM PDT
Impact	Unknown
ECID	Unknown
Data Dumped	No
Source	System Generated
Correlation Keys	SID = 129.232, ProcId = 39.24 PQ = (16777216, 1175542056), Client ProcId = oracle@stacg17 (TNS V1-V3).15201_3083220672
Purge Date	April 22, 2007 5:30:24 AM PDT (Purging Enabled) Disable Purging

[Dump Files](#) [Checker Findings](#) [Additional Diagnostics](#)

Search

Description	Damage Translation	Status	Time Detected
		Open	All

Data Corruption

Select findings and click on the "Launch Recovery Advisor" button to repair those findings.

[Launch Recovery Advisor](#)

[Select All](#) | [Select None](#) | [Expand All](#) | [Collapse All](#)

Select Description	Priority	Damage Translation	Incident ID	Status	Time Detected
<input type="checkbox"/> All Findings					
<input type="checkbox"/> Datafile 1: '/ade/ainc_emdbsa_b/oracle/dbs/t_db1.f'	High	Some objects in tablespace SYSTEM might be unavailable	3953	Open	April 2, 2007 12:27:41 PM PDT

ORACLE

Copyright © 2010, Oracle. All rights reserved.

서비스 요청 추적 및 복구 구현(계속)

Incident Details 페이지에서는 Oracle Advisor를 실행하여 복구를 구현할 수 있습니다.

다음 중 한 위치에서 권장되는 Advisor에 액세스합니다.

- Problem Details 페이지의 "Investigate and Resolve" 섹션에 있는 Self-Service 탭을 통해 액세스
- 슬라이드에 나와 있는 Incident Details 페이지의 Checker Findings 하위 페이지

Critical 오류 복구를 도와주는 Advisor는 다음과 같습니다.

- **Data Recovery Advisor:** 손상된 블록, 손상되었거나 누락된 파일 및 기타 데이터 Failure
- **SQL Repair Advisor:** SQL statement failure

Incident 및 문제 닫기

The screenshot shows the Oracle Support Workbench interface. At the top, it displays 'Problem Details: ORA 1578' and the date 'Page Refreshed July 9, 2007 12:53:11 AM GMT+07:00'. On the left, there's a 'Summary' section with fields like SR#, Bug#, Active (Yes), Packaged (No), and Number of Incidents (1). Below that is the 'Last Incident' section with details such as Timestamp (July 9, 2007 12:43:48 AM GMT+07:00), Incident Source (System Generated), Impact, Checkers Run (1), and Checker Findings (1). To the right, there are two 'Confirmation' boxes. The top one asks 'Are you sure you want to close the problem: ORA 1578?' with 'No' and 'Yes' buttons. The bottom one asks 'Are you sure you want to close the incident: 30177?' with 'No' and 'Yes' buttons. A red arrow points from the 'Close' button in the 'Last Incident' section to the 'Yes' button in the bottom confirmation box. Another red box highlights the 'Close' button in the 'Last Incident' section. The bottom of the screen features a red bar with the 'ORACLE' logo and the text 'Copyright © 2010, Oracle. All rights reserved.'

Incident 및 문제 닫기

특정 Incident를 더 이상 확인하지 않아도 되는 경우에는 Incident를 닫을 수 있습니다. 기본적으로 닫힌 Incident는 Problem Details 페이지에 표시되지 않습니다. 그리고 닫힘 여부에 관계없이 모든 Incident는 30일이 지나면 지워집니다. Incident Details 페이지에서 Incident 지우기를 비활성화할 수 있습니다.

Incident를 닫으려면 다음과 같이 하십시오.

1. Support Workbench 홈 페이지에 액세스합니다.
2. 원하는 문제를 선택하고 View를 누릅니다.
Problem Details 페이지가 나타납니다.
3. 닫을 Incident를 선택하고 Close를 누릅니다.
Confirmation 페이지가 나타납니다.
4. Confirmation 페이지에서 Yes를 눌러 Incident를 닫습니다.

Incident 패키지 구성

The screenshot shows the Oracle Support Workbench interface. On the left, there's a sidebar with 'Support Workbench' and 'Problems (4)'. Below it is a table showing 'New Problems in Last 24 Hour' and 'New Incidents in Last 24 Hour'. The main area displays the 'Edit Incident Packaging Configuration' dialog. This dialog has two tabs: 'Incident Data Retention' and 'Packaging Settings'. The 'Incident Data Retention' tab shows settings for 'Incident Metadata Retention Period (day)' (365) and 'Incident Files Retention Period (day)' (30). The 'Packaging Settings' tab shows settings for 'Cutoff Age for Incident Inclusion (day)' (90), 'Leading Incidents Count' (3), 'Trailing Incidents Count' (3), 'Correlation Time Proximity (min)' (90), and 'Time Window for Package Content (min)' (24). At the bottom of the dialog are 'Edit' and 'OK' buttons. A red box highlights the 'Edit' button. To the right of the dialog is a sidebar with '06 AM PDT Refresh' and 'Edit OK' buttons. A red arrow points from the 'Edit' button in the dialog to the 'Edit' button in the sidebar. Below the dialog is a 'Related Links' section with 'Alert Log Contents', 'Incident Packaging Configuration' (which is highlighted with a red box), and 'Alert Log Errors'. A red arrow points from the 'Incident Packaging Configuration' link to the 'Edit' button in the sidebar.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Incident 패키지 구성

Retention 규칙 및 패키지 생성을 구성할 수 있습니다. Support Workbench 페이지의 Related Links 섹션에서 Incident Package Configuration 링크를 눌러 Incident Packaging Configuration 페이지에 액세스합니다. 여기서 다음과 같은 파라미터를 변경할 수 있습니다.

- **Incident Metadata Retention Period:** 메타 데이터는 데이터에 대한 정보입니다. 그리고 Incident의 경우 메타 데이터에는 Incident 시간, ID, 크기 및 문제가 포함됩니다. 데이터는 Trace 등 Incident의 실제 내용입니다.
- **Cutoff Age for Incident Inclusion:** 이 값에는 현재까지의 범위에 속하는 패키지의 Incident가 포함됩니다. 차단 날짜를 90으로 설정하면 시스템에는 지난 90일 이내에 발생한 Incident만 포함됩니다.
- **Leading Incidents Count and Trailing Incidents Count:** 패키지에 포함된 모든 문제에 대해 시스템에서는 문제의 시작(선행) 및 종료(후행)에 있는 일정한 수의 Incident를 선택합니다. 예를 들어, 특정 문제에 Incident가 30개 있는 경우 선행 Incident 수가 5이고 후행 Incident 수가 4이면 시스템에서는 처음 5개의 Incident와 마지막 4개의 Incident를 포함합니다.
- **Correlation Time Proximity:** 이 파라미터는 "동시에 발생한 Incident"를 정의하는 시간 간격입니다. 다른 Incident 또는 문제와 상관되는 Incident 또는 문제를 참고하면 "서로 관련된 문제"를 손쉽게 파악할 수 있습니다. 상관 관계의 기준 중 하나는 시간 상관 관계입니다. 즉, 특정 문제의 Incident와 동시에 발생한 Incident를 찾는 것입니다.

ASM용 Enterprise Manager Support Workbench

The screenshot shows the Oracle Enterprise Manager 11g Support Workbench interface. At the top, it displays 'ORACLE Enterprise Manager 11g Database Control' and the URL 'Automatic Storage Management +ASM edrsr22p1.us.oracle.com'. Below this, the title 'Support Workbench' is shown, along with a timestamp 'Page Refreshed June 25, 2009 8:5'. A navigation bar at the top includes tabs for 'Problems (0)', 'Checker Findings (0)', and 'Packages (0)'. The main content area shows statistics for 'New Problems in Last 24 Hours' (0), 'All Active Problems' (0), 'All Problems' (0), 'New Incidents in Last 24 Hours' (0), 'All Active Incidents' (0), and 'All Incidents' (0). There is a search bar and a dropdown menu set to 'Last 24 Hours'. A table below lists 'No problems found.' under the 'ID Description' column. Another navigation bar at the bottom includes tabs for 'Problems (0)', 'Checker Findings (0)', and 'Packages (0)'. A 'Related Links' section at the bottom right includes links to 'Alert Log Contents', 'Incident Packaging Configuration', 'Alert Log Errors', 'Checker Central', 'Create User-Reported Problem', and 'Support Workbench (orcl)'. The bottom of the page features a red footer bar with the 'ORACLE' logo and the text 'Copyright © 2010, Oracle. All rights reserved.'

ASM용 Enterprise Manager Support Workbench

Oracle Enterprise Manager는 진단을 돋기 위해 성능이 향상되었으며 Oracle ASM Instance에 대한 Oracle 지원 센터 서비스로 Incident를 패키지화하여 제출합니다.

Oracle Enterprise Manager는 Oracle ASM Alerts 및 Incident를 모니터하기 위한 Oracle ASM Support Workbench를 제공합니다.

Oracle ASM에 대한 Support Workbench에 액세스하려면:

1. Database Home 페이지에서 Software and Support 탭을 누릅니다.
2. Software and Support 페이지의 Support 섹션에서 Support Workbench를 누릅니다.
3. Support Workbench 페이지의 Related Links 섹션 아래에서 Support Workbench(*ASM_instance_name*)를 누릅니다.

Problems 페이지에서 현재 및 이전 문제에 대한 정보를 볼 수 있습니다.

Oracle 지원 센터 서비스에 보낼 패키지를 생성하려면 Support Workbench Problems 페이지에서 Incident를 선택하고 Package를 누릅니다. 그런 다음 Support Workbench 안내에 따라 패키지화 프로세스를 진행할 수 있습니다.

Oracle 지원 센터 활용

- OSS(Oracle 지원 센터 서비스)는 연중 무휴로 솔루션 지원을 제공합니다.
- 지원은 다음과 같은 방식으로 제공됩니다.
 - My Oracle Support 웹 사이트
 - 전화
 - ODC(Oracle Direct Connect) 원격 진단 도구
- 각 고객에게 사용권이 있는 소프트웨어 및 지원을 추적할 때는 CSI(Customer Support Identifier) 번호가 사용됩니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle 지원 센터 활용

OSS(Oracle 지원 센터 서비스)는 전 세계의 모든 오라클 고객에 대해 연중 무휴로 솔루션 지원을 제공합니다. OSS는 필요할 때는 언제든지 이러한 지원을 제공하기 위한 지원 센터를 전 세계에 보유하고 있습니다.

오라클 고객에 대한 지원은 My Oracle Support 웹 사이트, 전화 또는 ODC(Oracle Direct Connect) 원격 진단 도구를 통해 제공됩니다.

Oracle 소프트웨어를 구입하는 고객에게는 CSI(Customer Support Identifier) 번호가 제공됩니다. 이 번호는 각 고객에게 사용권이 있는 소프트웨어 및 지원을 추적하는 데 사용됩니다. CSI 번호를 사용하면 My Oracle Support에서 제공되는 모든 패치, 설명서 및 문제 해결 정보에 액세스할 수 있으며, OSS에 대해 SR(서비스 요청)을 로깅할 수 있습니다.

주: 서비스 요청은 예전의 TAR(Technical Assistance Request)입니다.

My Oracle Support 통합

- Enterprise Manager는 새 중요 패치에 대한 알림을 유저에게 자동으로 제공합니다.
- Enterprise Manager 패치 마법사를 사용하여 Interim 패치를 선택할 수 있습니다.
- Enterprise Manager 내에서 패치의 README 파일을 직접 확인할 수 있습니다.
- My Oracle Support에서 선택한 패치를 Enterprise Manager 패치 캐시로 다운로드할 수 있습니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

My Oracle Support 통합

Oracle Enterprise Manager(간단히 Enterprise Manager)는 내장된 My Oracle Support 통합을 사용하여 소프트웨어 패치 작업을 손쉽게 처리합니다. Enterprise Manager는 새 중요 패치에 대한 알림을 유저에게 자동으로 제공하며 특정 패치를 필요로 하는 모든 시스템에 플래그를 설정합니다.

Enterprise Manager 패치 마법사를 호출하여 설치 가능한 Interim 패치를 확인할 수 있으며, 패치 마법사를 통해 Interim 패치를 선택하여 이 패치를 필요로 하는 시스템을 확인할 수도 있습니다. Enterprise Manager 내에서 직접 패치 세부 정보와 README 패치 노트를 확인할 수 있습니다.

Enterprise Manager 패치 마법사를 사용하여 My Oracle Support에서 Interim 패치를 Enterprise Manager 패치 캐시로 다운로드하면 다운로드를 반복할 필요가 없게 됩니다. 대상 시스템에 적절한 패치를 준비해 두면 나중에 수동으로 적용할 수 있습니다. 패치 프로세스를 더 많이 자동화하려면 제공된 Enterprise Manager 에이전트에서 유저가 정의한 시간에 대상 시스템에 대해 실행되는 패치 적용 스크립트를 커스터마이즈할 수도 있습니다. 패치를 시스템에 적용하면 시스템의 정확한 패치 레벨을 추적하기 위해 해당 OUI(Oracle Universal Installer) 인벤토리가 자동으로 갱신됩니다.

Maintenance 페이지의 Deployments 영역에서 Patch를 눌러 패치 마법사에 액세스합니다.

My Oracle Support 사용

The screenshot shows the Oracle My Oracle Support dashboard. At the top, there's a banner with a video tutorial link and a configuration manager note. Below the banner, there are several panels: 'System Health' (with a 'Getting Started' guide), 'Service Requests' (empty), 'Systems' (empty), 'Targets' (empty), 'News' (listing recent articles like 'Classic MetaLink to be Retired'), and 'Knowledge Articles' (listing alerts like 'About Oracle Customer Support Minipack 11i.CSZ.A'). A 'Select Platform' dropdown is set to 'Linux x86'. The bottom of the page has a red footer with the Oracle logo and copyright information.

My Oracle Support 사용

My Oracle Support를 등록하려면 <http://myoraclesupport.oracle.com/>으로 이동하고 Register here 링크를 선택합니다. 프롬프트가 표시되면 CSI 번호를 입력하고 몇 가지 간단한 질문에 대한 답변을 입력합니다. 등록이 완료되면 My Oracle Support를 사용할 수 있습니다. 각 CSI 번호에는 My Oracle Support에 대한 새 유저 액세스를 제어하는 고객이 지정한 관리자가 있습니다. 고객은 이 개인 관리자를 지정해야 하며, 그러면 새 유저는 이 개인 관리자를 통해 새 계정을 생성하고 적절한 My Oracle Support 액세스를 부여해야 합니다.

My Oracle Support에는 문제 조사에 사용할 수 있는 다양한 도구 및 방법이 있습니다.

My Oracle Support에서 표준 검색 엔진 및 고급 검색 엔진을 통해 답변을 검색하는 작업은 비교적 간단합니다. 일반적인 문제에 대해서는 너무 많은 결과가 반환될 수 있습니다. 다음과 같은 몇 가지 간단한 단계를 수행하면 검색 결과의 정확도와 관련성을 높일 수 있습니다.

- 검색을 수행할 때는 정확한 전체 오류 텍스트를 사용하십시오. 예를 들어, ORA-1400 : mandatory (NOT NULL) column을 입력하면 ORA-1400보다 관련성이 높은 답변이 반환됩니다.
- Oracle E-Business Suite에서 오류를 조사할 때는 검색 조건의 일부로 코드 이름을 입력하십시오. 예를 들어, 오류 메시지만 입력하는 것보다는 APXINWKB ORA-1400 : mandatory (NOT NULL) column이라고 입력하면 보다 적은 수의 적절한 결과가 반환됩니다.

My Oracle Support 사용(계속)

키워드를 검색하지 않고 정보를 검색하는 드릴 다운 방식을 사용하려는 경우에는 Knowledge 탭을 사용하여 Knowledge Browser에 액세스하면 됩니다. Knowledge Browser에서는 OSS에서 가장 자주 사용되는 기술 컨텐트에 쉽게 액세스할 수 있습니다.

Knowledge Browser는 다음과 같은 최신 정보를 쉽게 제공할 수 있도록 구성되었습니다.

- *Featured News and Articles* 섹션의 최근 발표 내용 및 정보
- 제품 범주별 정보
- 사례 연구
- 도구 및 교육
- 온라인 설명서
- eTRM(전자 기술 참조 설명서)
- Oracle Integration Repository
- Customer Knowledge Exchange

My Oracle Support 포럼(간단히 포럼)에서는 다른 오라클 고객과의 상호 작용을 통해 의견을 교환하고 오라클 제품에 대해 토론을 할 수 있습니다. 또한 My Oracle Support 포럼에서는 다른 고객들이 복잡한 작업을 수행하는 방법이나 오라클 제품을 통해 다양한 업무 요구 사항을 충족하는 방식을 확인할 수 있습니다. 포럼은 SR 로깅 대용으로 사용해서는 안됩니다.

고객은 패치 엔진을 사용하여 다양한 방식으로 패치를 검색할 수 있습니다. 그 중 가장 일반적으로 사용되는 패치 검색 방법은 다음과 같습니다.

- **패치 번호:** 패치 번호를 아는 경우 입력하면 됩니다.
- **마지막으로 통합된 패치:** 업그레이드 시 사용 중인 제품의 최신 패치를 확인하려는 경우 이 방법을 사용하면 됩니다.
- **파일 포함:** 특정 코드 부분에서 문제가 발견되면 일반적으로는 패치를 사용하여 문제를 해결할 수 있습니다. 그러므로 지원 담당자는 보통 고객에게 패치를 적용해 해당하는 릴리스에 대해 사용 가능한 최신 버전으로 코드를 갱신할 것을 권장합니다. 코드의 이름 및 버전을 확인한 다음 패치 검색 유ти리티를 사용하여 해당 코드에 대해 보다 최신 버전이 사용 가능한지를 확인함으로써 최신 버전의 Oracle 소프트웨어를 찾아 적용할 수 있습니다.

주: 이러한 검색을 수행하는 방법에 대한 자세한 사항은 My Oracle Support Technical Note 166650.1("Working Effectively with Global Customer Support")을 참조하십시오.

BUGs 링크를 사용하면 문제를 조사할 때 버그 데이터베이스를 검색할 수 있습니다. 다양한 방법을 사용하여 버그 데이터베이스를 검색할 수 있습니다.

문제 조사

My Oracle Support에서 문제를 조사하려면 다음 단계를 수행하십시오.

- 1. 키워드 검색을 수행합니다.**
- 2. 설명서를 검토합니다.**
- 3. 셀프 서비스 툴킷을 사용합니다.**
- 4. 자동화된 진단 테스트 및 업무 흐름을 사용합니다.**
- 5. 적용 가능한 패치를 검색합니다.**
- 6. SR(서비스 요청)을 로깅합니다.**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

문제 조사

My Oracle Support에서는 문제 조사에 사용할 수 있는 다양한 리소스를 제공합니다.

다음 단계에서는 My Oracle Support 리소스를 사용하는 기본적인 문제 해결 기술에 대해 간략하게 설명합니다.

- 1. 키워드 검색:** 대부분의 문제는 My Oracle Support에서 키워드 검색 유ти리티를 사용하면 빠르고 쉽게 해결할 수 있습니다. 효율적인 검색을 수행하면 특정 문제 및 해결 방법에 대한 많은 정보를 확인할 수 있습니다.
- 2. 설명서:** 키워드 검색을 통해 해결 방법을 찾지 못한 경우에는 설치 과정의 문제가 근본적인 원인이 아닌지 설명서를 검토해야 합니다. 설치 문제는 모든 서비스 요청의 1/3 이상을 차지하므로, 문제 해결 프로세스에서 초기에 설치 과정을 검토하는 것이 좋습니다. 설명서는 PDF 형식으로 게시된 유저 설명서 및 구현 설명서를 비롯하여 HTML로 게시된 제품 README 파일과 설치 노트로 구성됩니다. 이 두 가지 문서 유형은 모두 My Oracle Support에서 제공되며, 각 제품의 셀프 서비스 툴킷을 통해서도 액세스할 수 있습니다.

문제 조사(계속)

3. **셀프 서비스 툴킷:** SSTK(셀프 서비스 툴킷)는 각 제품에 대한 다양한 정보를 제공합니다. 대부분의 경우 이러한 툴킷에는 FAQ, 패치 리스트 및 발생한 문제를 조사하고 해결할 수 있도록 도와주는 기타 유용한 정보가 포함되어 있습니다. SSTK에는 각 제품과 관련하여 가장 자주 사용되는 컨텐트가 들어 있으므로, 작업 중인 환경 내에서 문제가 발생하기 전에 알려진 문제를 파악할 수 있도록 이러한 컨텐트를 주기적으로 참조해야 합니다.
4. **진단 및 흐름:** Oracle 지원 센터 서비스에서 최근에 개선된 내용은 대부분 자동화된 진단 테스트 및 업무 흐름 분야에 집중되어 있습니다. 즉, 시스템 설치를 확인하거나 문제에 대한 정보를 수집할 수 있도록 테스트 및 흐름이 새롭게 생성되었습니다. 진단 테스트의 경우에는 Java 또는 SQL 스크립트를 실행하여 이를 수행할 수 있습니다. 이러한 테스트의 결과는 문제를 해결하는 데 도움이 될 뿐 아니라 서비스 요청을 로깅해야 하는 경우 Oracle 지원 센터 서비스에서 문제의 원인을 파악하는 데도 도움이 됩니다.
5. **패치 및 버그:** 오라클 제품에서 버그가 발견되면 패치를 사용해 문제를 해결해야 합니다. 문제를 해결할 때는 보다 최신 제품 릴리스를 제공하는 패치를 사용할 수 있는지 여부를 확인하기 위해 시스템을 검토해야 합니다. 패치 검색 도구를 사용하면 특정 파일이 포함된 패치를 검색할 수 있습니다. 최신 코드를 검색하여 환경에 최신 버전의 패치를 적용하면 문제의 원인이 될 수 있는 기존 버그를 제거함으로써 문제 해결 프로세스를 개선할 수 있습니다. 또한 버그 검색 엔진을 활용하여 문제에 대해 로깅되었지만 아직 해결되지 않은 버그가 있는지를 확인해야 합니다.
6. **SR(서비스 요청) 로깅:** 셀프 서비스 옵션을 모두 사용했지만 해결 방법을 찾지 못한 경우에는 지원 담당자에게 문제 해결 지원을 요청해야 할 수 있습니다.

서비스 요청 로깅

- **My Oracle Support 홈 페이지의 Service Request 탭을 눌러 SR을 로깅합니다.**
- **My Oracle Support는 CSI 번호 및 SR 프로파일을 기반으로 검색을 수행합니다.**
- **SR을 로깅할 때는 다음 정보를 제공하십시오.**
 - 문제 설명(오류 메시지 포함)
 - 문제 해결을 위해 수행한 단계
 - 소프트웨어 버전
 - 문제 재현에 필요한 단계
 - 문제가 업무에 미치는 영향

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

서비스 요청 로깅

My Oracle Support에서 문제에 대해 조사해도 해결 방법을 찾을 수 없는 경우가 있습니다. 이 경우에는 My Oracle Support를 통해 SR(서비스 요청)을 로깅해야 합니다. My Oracle Support 홈 페이지에서 Service Request 탭을 누르면 SR을 로깅할 수 있습니다.

SR을 생성할 때 첫번째로 수행하는 단계는 CSI 번호 및 SR 프로파일을 선택하는 것입니다. 필요한 프로파일 정보를 제출(Submit)하고 나면 My Oracle Support가 문제 유형, 오류 메시지, 문제에 대한 간략한 요약 및 언어 환경 설정과 같은 문제와 관련된 일부 구체적인 사항을 수집합니다. My Oracle Support는 이 정보를 사용하여 검색을 수행하고 해결 방법을 찾으려고 시도합니다.

이 단계 중에 수행되는 검색을 통해 앞서 수행한 검색과 다른 결과를 얻을 수도 있습니다. 두 검색에서는 모두 동일한 데이터베이스에서 정보 및 버그를 검색하지만, 검색 엔진과 가중치가 약간 다르기 때문입니다. 검색 결과가 달라질 수 있으므로 My Oracle Support 검색 엔진을 사용하여 이전 검색을 수행한 경우에도 SR 생성 프로세스 중에 검색 결과를 검토해야 합니다.

서비스 요청 로깅(계속)

검색 결과를 통해서도 문제를 해결하지 못하면 SR 생성 프로세스에서 계속해서 일련의 질문을 하고 정보를 요청합니다. 질문에 대답하고 나면 SR이 전자 방식으로 제출(Submit)되어 문제를 자세하게 분석하는 지원 담당자에게 전달됩니다. SR이 로깅된 후에는 My Oracle Support의 SR 섹션에서 제공되는 업로드 유ти리티를 사용하여 모든 파일, 스크린샷 및 기타 추가 정보를 즉시 업로드해야 합니다.

SR에 다음과 같은 항목이 명확하게 제시되어 있는지 확인해야 합니다. 이러한 정보를 제공하면 지원 담당자가 해당 문제를 우선적으로 확인하고 작업을 처리할 수 있습니다.

- 문제에 대한 명확한 설명(정확한 오류 메시지 포함)
- 문제를 해결하기 위해 수행한 단계 및 해당 결과에 대한 설명
- 정확한 소프트웨어 버전
- 문제 재현에 필요한 단계
- 중요 시점, 날짜, 비용 등 해당 문제가 업무에 미치는 영향

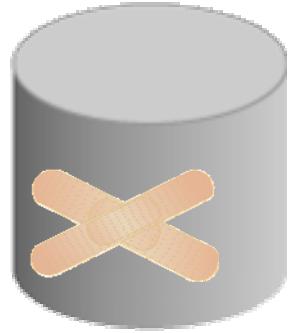
각 SR에는 *SR 번호*라는 고유한 식별자가 할당됩니다. SR을 로깅하면 My Oracle Support에서 SR 번호를 제공합니다. 전화로 SR을 로깅하는 경우에는 지원 담당자가 SR 번호를 알려 줍니다. 그런 후에 지원 담당자는 Oracle 지원 센터 서비스에서 모든 전화 및 웹 접수 서비스 요청을 분산하는 데 사용하는 자동화된 할당 프로세스를 통해 자신의 큐에서 SR을 받게 됩니다. 이와 같이 자동화된 프로세스를 사용하면 모든 SR이 보고되는 특정 문제를 가장 효율적으로 처리할 수 있는 지원 담당자에게 할당됩니다.

주: 자세한 사항은 My Oracle Support Technical Note 166650.1("Working Effectively with Global Customer Support")을 참조하십시오.

패치 관리

패치 종류

- **Interim 패치**
 - 특정 문제에 사용
 - 회귀 테스트 없음
- **CPU(Critical Patch Update)**
 - 중요한 보안 문제
 - 회귀 테스트
 - 버전 번호를 제출하지 않음
- **패치 릴리스**



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

패치 관리

다양한 원인으로 인해 각 패치 종류를 서로 다른 시간에 적용할 수 있습니다.

- Interim 패치(일회성 패치 또는 단일 패치라고도 함)는 특정 문제를 해결하기 위해 생성되며, 전체 회귀 테스트를 거치지 않습니다. Interim 패치는 보통 opatch 유ти리티와 함께 설치됩니다. Enterprise Manager Patching Wizard를 통해 패치를 다운로드, 적용 및 스테이지함으로써 패치 적용 프로세스를 손쉽게 자동화할 수 있습니다. 이 마법사는 백그라운드에서 opatch 유ти리티를 사용합니다.
- CPU(Critical Patch Update) 패치에는 보안 패치 및 종속되는 비보안 패치가 포함됩니다. CPU 패치는 누적 패치이므로 이전 Oracle 보안 Alert의 수정 사항과 중요 패치 갱신을 포함합니다. CPU 패치를 적용하기 전에 이전 보안 패치를 적용할 필요는 없습니다. 그러나 지정된 패치 세트 레벨에서 적용해야 합니다. CPU 패치는 10.2.0.3 등 특정 패치 릴리스 레벨에 적용하기 위한 것이며, opatch 유ти리티 또는 EM Patching Wizard를 통해 설치됩니다. CPU 패치는 분기별로 발행됩니다. CPU 패치 및 Interim 패치는 opatch rollback -id <patch id>를 사용하여 시스템에서 제거할 수도 있습니다. 오라클은 오라클 고유 응용 프로그램을 사용하여 Critical Patch Update에 대한 광범위한 테스트를 실행함과 동시에 Critical Patch Updates 자체에 대해서도 회귀 테스트를 실행합니다. 패치가 적용되었는지 확인하려면 opatch -lsinventory를 사용하여 인벤토리를 query한 후에 패치가 리스트에 있는지를 확인하십시오.

패치 릴리스 적용

- **패치 릴리스는 완전하게 테스트를 거친 다음과 같은 특징의 제품 수정 사항입니다.**
 - 새 기능을 포함하지 않음
 - 설치 시에 Oracle 홈에 상주하는 소프트웨어에만 영향을 줌
 - 개별 버그 수정 사항 포함
 - 버전 번호 적용
- **패치를 적용하려면 다음과 같이 하십시오.**
 1. Oracle 소프트웨어 환경을 확인합니다.
 2. My Oracle Support 로그인 인증서를 설정합니다.
 3. 패치 릴리스를 스테이지합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

패치 릴리스 적용

소프트웨어 관리에는 최신 제품 수정 사항을 적용하여 Oracle 소프트웨어를 최신 상태로 유지하는 작업이 포함됩니다. 오라클에서는 주기적으로 해당 소프트웨어용 패치 릴리스(제품 수정 사항)를 제공합니다. 패치 릴리스는 완전하게 테스트된 제품 수정 사항일 뿐이며, 새 기능을 포함하지는 않습니다. 패치 릴리스를 적용하면 Oracle 홈에 있는 소프트웨어에만 영향을 주며 데이터베이스는 업그레이드되거나 변경되지 않습니다.

패치는 개별 버그 수정 사항이며, 패치 세트는 릴리스 시까지의 버그 수정 사항이 포함된 컬렉션입니다. 모든 패치 및 패치 세트 릴리스에는 버전 번호가 적용됩니다. 예를 들어, Oracle Database 11g Release 11.1.0.2를 구입한 경우 사용 가능한 패치 세트는 11.1.0.3입니다. 모든 패치 또는 패치 세트에는 식별을 위한 패치 번호도 포함되어 있습니다. 그리고 모든 패치 릴리스에는 해당 버그 수정 사항을 설명하는 README 파일이 연관되어 있습니다. README에는 패치를 수동으로 적용하기 위한 지침도 포함되어 있습니다.

Enterprise Manager를 사용하면 My Oracle Support 웹 사이트에서 최신 패치 릴리스를 찾아 Oracle 홈으로 다운로드할 수 있습니다.

Patch Advisor 사용

The screenshot shows the Oracle Patch Advisor interface. At the top, it says "Logged In As: [User]". Below that, there's a section titled "Critical Security Patches" with a sub-section "Patch Recommendations by Feature". It includes a dropdown menu "View Based on Usage" with options "Go", "(Schedule Patching)", "Select All", and "Select None". A table lists three patches:

Select Patch Number	Created On	Description	Impacted Feature	README
<input type="checkbox"/> 4751921	2007-02-14	A useful Patch	Services	View
<input type="checkbox"/> 4751923	2007-02-15	Another useful patch	Services, Data Mining	View
<input type="checkbox"/> 4751925	2007-02-15	Yet Another useful patch	Audit Options	View

A note at the bottom of this section says "TIP It is recommended to check patch prerequisites before applying patches." Below this, there's a "Related Links" section with links to "Patch Prerequisites", "Database Feature Usage", "Interim Patches Applied", "Stage Patch", and "Patching Setup".

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Patch Advisor 사용

Patch Advisor에는 시스템에 적용할 수 있는 Critical Patch Update와 권장 패치가 표시됩니다. 사용하는 기능에 따라 권장 패치를 표시할 수도 있고 사용 가능한 모든 패치를 표시할 수도 있습니다. 패치를 표시하려면 패치 설치를 구성하고 RefreshFromMetalink 작업을 실행해야 합니다.

Patch Advisor 페이지의 Related Links 섹션에서 Patch Setup을 누르면 Patch Setup 페이지로 이동합니다.

Patch Wizard 사용

Select the Patches to apply. Click on "Add Patches" to search and select patches from Metalink or Software Library.

Target List

Instance Name	database
Target Type	Database Instance
Release	11.1.0.4.0
Host	std129.us.oracle.com
Staging Location	%oracle_home%/EMStagedPatches

This is the directory on the host where the updates will be staged.

Patches

Add Patches	Software Update Name	Patch Number	Created On	Type	Product	Platform	Release	Interim Patch Applicable On	Description
	p4751921_11.1.0.4.0_46_9480	4751921	2007-02-14 00:00:00.0	Patch	Oracle Database	Linux x86	11.1.0.4.0		A useful Patch

Post Patch SQL to apply

Default (for Critical Patch Updates and Patchsets)

Custom SQL File Path Specify the file location on the host (e.g., %oracle_home%/files/patch.sql).

None

Step 1 of 5

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Patch Wizard 사용

"Software and Support" 페이지의 Database Software Patching 섹션에서 Apply Patch를 누르면 Patch Wizard가 호출됩니다.

여기서 첫번째로 수행하는 단계는 Select Patches입니다. Add Patch를 눌러 이 일정이 잡힌 실행에서 적용할 패치를 추가로 선택합니다.

Target List 단계는 RAC 패치용으로 예약되는 단계이므로 Oracle Database 11g에서는 건너뜁니다. 고객이 커스텀 변수를 사용하여 배치 프로시저를 커스터마이즈한 경우가 아니면 Library Step Properties도 건너뜁니다. 고객이 배치 프로시저를 커스터마이즈한 경우에는 Library Step Properties를 건너뛰지 않으며, 유저가 커스텀 변수의 값을 입력합니다.

다음 단계에서는 패치 작업을 실행하고 작업을 즉시 실행할지 나중에 실행할지를 결정하기 위한 인증서를 제공합니다.

그런 후에 작업을 검토 및 제출(Submit)하면 됩니다.

패치 적용

Search And Select Patches

Search Metalink
 Search Software Library

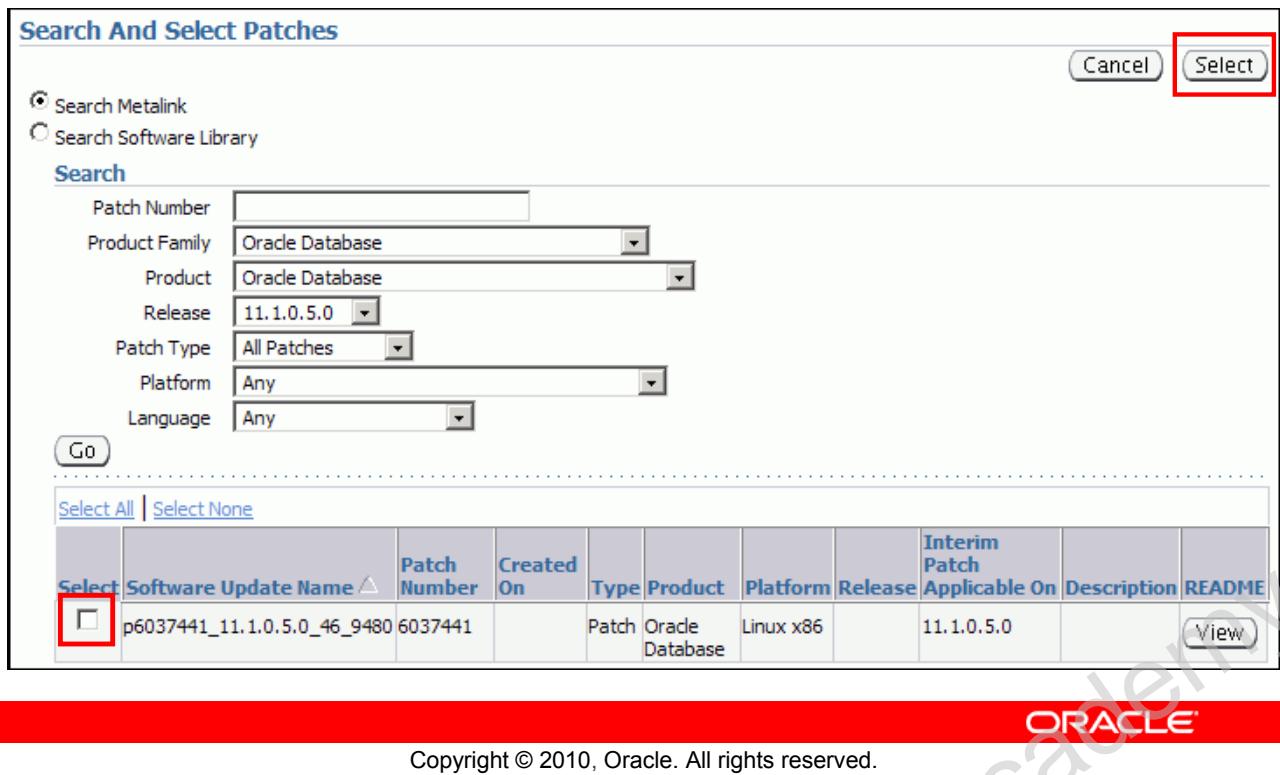
Search

Patch Number	<input type="text"/>
Product Family	Oracle Database
Product	Oracle Database
Release	11.1.0.5.0
Patch Type	All Patches
Platform	Any
Language	Any

[Select All](#) | [Select None](#)

Select	Software Update Name ▲	Patch Number	Created On	Type	Product	Platform	Release	Interim Patch Applicable On	Description	README
<input type="checkbox"/>	p6037441_11.1.0.5.0_46_9480	6037441		Patch	Oracle Database	Linux x86		11.1.0.5.0		<input type="button" value="View"/>

ORACLE
Copyright © 2010, Oracle. All rights reserved.



패치 적용

"Software and Support" 페이지를 사용하면 패치, CPU 또는 패치 릴리스를 찾아 적용할 수 있습니다.

패치 스테이지

To look up patches at OracleMetaLink, enter search criteria and click Search. This may take a few moments depending on the number of matches found. From Search Results, select the patch to be applied and click Next.
For advanced features like multiple patch application, patch flow customization, sudo and PAM support please use the " [Deployment Procedures](#)" functionality. For details on Deployment Procedures, consult the relevant [documentation](#)

Search by Number

* Patch Number [Search by Criteria](#)

Platform (If you run a 32-bit Oracle product on a 64-bit operating system, choose a 32-bit platform.)

Language

[Search](#) [Clear](#)

Search Results

Select	Patch Number	Created On	Type	Product	Platform	Release	Interim Patch Applicable On	Description	Status
<input checked="" type="radio"/>	6037441	May 11, 2007	Patch	Oracle Database	Linux x86		11.1.0.5.0	DUMMY BUG FOR DUMMY PATCH UPLOAD TO ARU FOR BETA TESTING	available

ORACLE

Copyright © 2010, Oracle. All rights reserved.

패치 스테이지

"Software and Support" 페이지의 Database Software Patching 섹션에서 Stage Patch를 누르면 Patch Wizard가 호출됩니다.

첫번째로 수행하는 단계는 번호 또는 기준별로 패치를 선택하는 것입니다.

그런 다음 대상을 선택합니다. 이 단계에서는 사용 가능한 대상 리스트에서 원하는 항목을 선택할 수 있습니다.

세번째 단계에서는 패치 적용을 수행하는 OS 유저의 인증서를 제공합니다. 이 유저를 소프트웨어 설치를 소유한 유저로 지정할 것을 권장합니다.

다음 단계에서는 패치를 스테이지할 것인지 스테이지 후 적용할 것인지를 선택할 수 있습니다.

다섯번째 단계에서는 작업의 일정을 잡습니다.

마지막 단계에서는 패치 작업을 검토 및 제출(Submit)할 수 있습니다.

스테이지된 패치는 UNIX 및 Linux 플랫폼의 경우 \$ORACLE_HOME/EMStagedPatches_<sid> 디렉토리에 저장되고, Windows 플랫폼의 경우에 %ORACLE_HOME%\EMStagedPatches_<sid> 디렉토리에 저장됩니다.

온라인 패치: 개요

Oracle Instance에서 실행 중인 버그 수정 사항 또는 진단 패치의 경우 온라인 패치를 사용하면 다음이 가능합니다.

- 설치
- 활성화
- 비활성화



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

온라인 패치: 개요

온라인 패치를 사용하면 실행 중인 실제 Oracle Instance에서 버그 수정 사항 또는 진단 패치를 설치, 활성화 및 비활성화할 수 있습니다. 온라인 패치 사용은 온라인 패치를 적용할 때 다운타임을 방지하기 위해 권장되는 해결책입니다. 오라클 데이터베이스에서 온라인 패치를 수행하려면 opatch 명령행 유ти리티를 사용합니다. 온라인 패치는 변경된 코드의 범위가 좁고 복잡하지 않을 경우에 제공됩니다(예: 진단 패치 또는 작은 버그 수정 사항).

온라인 패치 설치

- 온라인 패치를 적용할 때는 Instance 종료, 오라클 Binary 재연결 또는 Instance 재시작이 필요하지 않습니다.
- OPatch를 사용하여 온라인 패치를 설치하고 제거할 수 있습니다.
- OPatch는 두 온라인 패치 간의 충돌뿐만 아니라 온라인 패치와 일반 패치 간의 충돌도 감지합니다.
- 패치가 온라인 패치인지 확인하려면:

```
opatch query -is_online_patch <patch location>
또는
opatch query <patch location> -all
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

온라인 패치 설치

일반 패치 메커니즘과 달리 온라인 패치를 적용할 때는 Instance를 종료하거나 재시작할 필요가 없습니다.

일반 패치와 마찬가지로 OPatch를 사용하여 온라인 패치를 설치할 수 있습니다.

다음 명령을 사용하여 패치가 온라인 패치인지 여부를 확인할 수 있습니다.

```
opatch query -is_online_patch <patch location> or
opatch query <patch location> -all
```

주: 패치된 코드는 동적/공유 라이브러리로 제공되며, 각 오라클 프로세스에 의해 메모리에 매핑됩니다.

온라인 패치의 이점

- **다운타임 및 업무 방해 없음**
- **매우 빠른 설치 및 제거 시간**
- **OPatch와 통합:**
 - 충돌 감지
 - 패치 인벤토리에 나열됨
 - RAC 환경에서 작동
- **Instance 종료 및 시작 간에 유지됨**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

온라인 패치의 이점

온라인 패치를 적용하는 동안 데이터베이스 Instance를 종료할 필요가 없습니다. 일반 패치와는 달리 온라인 패치는 설치 및 제거가 빠릅니다. 온라인 패치는 OPatch를 사용하기 때문에 OPatch를 사용하는 일반 패치의 모든 이점을 그대로 활용할 수 있습니다. 온라인 패치는 Instance를 종료하고 시작할 때 그대로 유지되므로 데이터베이스를 얼마나 오래 또는 얼마나 자주 종료하는지는 상관이 없습니다.

일반 패치 및 온라인 패치

일반 패치	온라인 패치
적용하거나 제거할 때 다운타임이 필요함	적용하거나 제거할 때 다운타임이 필요 없음
OPatch를 통해 설치 및 제거됨	OPatch를 통해 설치 및 제거됨
Instance 시작 및 종료 간에 유지됨	Instance 시작 및 종료 간에 유지됨
설치 또는 제거에 몇 분이 걸림	설치 또는 제거에 몇 초밖에 걸리지 않음

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

일반 패치 및 온라인 패치

일반 패치에서는 기본적으로 데이터베이스 Instance를 종료해야 합니다.

온라인 패치는 다운타임이 필요하지 않으므로 온라인 패치를 설치하는 동안에도 응용 프로그램을 계속 실행할 수 있습니다. 마찬가지로, 설치된 온라인 패치를 제거할 때도 다운타임이 필요하지 않습니다.

온라인 패치 고려 사항

- 온라인 패치는 다음과 같은 플랫폼에서 지원됩니다.
 - Linux x86 32/64
 - HP Itanium
 - Sun Sparc Solaris 64
 - AIX
 - Windows x86 32/64
- 일부 메모리가 추가로 소비됩니다.
 - 정확한 양은 다음에 따라 다릅니다.
 - 패치 크기
 - 동시에 실행 중인 오라클 프로세스의 수
 - 최소 메모리 양: 실행 중인 오라클 프로세스당 OS 페이지 하나 정도

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

온라인 패치 고려 사항

운영 체제(OS) 페이지 하나는 일반적으로 Linux x86에서 4KB, Solaris SPARC64에서는 8KB입니다. 동시에 실행 중인 오라클 프로세스의 평균 숫자가 약 천 개일 경우에는 작은 온라인 패치에 약 4MB의 추가 메모리가 사용되는 것입니다.

온라인 패치 고려 사항

- 모든 오라클 프로세스에서 온라인 패치를 설치하거나 제거할 때까지 약간의 지연 시간(몇 초)이 있을 수 있습니다.
- 모든 버그 수정 사항과 진단 패치가 온라인 패치로 제공되는 것은 아닙니다.
- 다운타임이 있어서는 안되는 경우에 온라인 패치를 사용합니다.
- 다운타임이 허용되는 경우에는 모든 관련 버그 수정 사항을 일반 패치로 설치해야 합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

온라인 패치 고려 사항(계속)

대부분의 진단 패치는 온라인 패치로 제공됩니다. 버그 수정 사항의 경우에는 해당 버그 수정 사항의 특징에 따라 다릅니다. 모든 버그 수정 사항이나 진단 패치가 온라인 패치로 제공되는 것은 아니지만 온라인 패치 담당 기관의 장기적인 목표는 주요 패치 개선에 대해 온라인 패치 기능을 제공하는 것입니다.

주: 일반 패치를 적용하기 전에 온라인 패치를 제거해야 합니다.

퀴즈

온라인 패치에 대한 다음 설명 중 옳은 것은?

1. OPatch를 사용하여 설치할 수 있음
2. 적용하려면 다운타임이 필요함
3. Instance 시작 및 종료 간에 유지됨
4. 제거할 때 다운타임이 필요 없음

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1, 3 및 4

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- Support Workbench 사용
- Oracle 지원 센터 활용
- My Oracle Support 검색
- 서비스 요청 로깅
- 패치 관리
 - 패치 릴리스 적용
 - 패치 릴리스 스테이지

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

연습 18 개요: Alert 및 패치에 EM 도구 사용

이 연습에서는 Support Workbench를 사용하여 Critical 오류를 조사하는 방법을 다룹니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

부록 A

연습 및 해답

Oracle Internal & Oracle Academy
Use Only

목차

단원 1 의 연습.....	4
연습 1-1: 오라클 데이터베이스 구조 탐색.....	5
단원 2 의 연습.....	9
연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성	10
연습 2-2: Oracle Database 11g 소프트웨어 설치	17
단원 3 의 연습.....	20
연습 3-1: 오라클 데이터베이스 생성	21
단원 4 의 연습.....	29
연습 4-1: Oracle Instance 관리.....	30
연습 4-2: Oracle Restart 및 데이터베이스 테스트	36
단원 5 의 연습.....	38
연습 5-1: ASM Fast Mirror Resync	39
연습 5-2: ASMCMD 사용	45
단원 6 의 연습.....	48
연습 6-1: 다른 데이터베이스를 액세스하기 위한 Oracle 네트워크 구성	49
연습 6-2: 대체 리스너 생성	54
단원 7 의 연습.....	57
연습 7-1: 데이터베이스 저장 영역 구조 정보 보기	58
연습 7-2: 테이블스페이스 생성	62
단원 8 의 연습.....	68
연습 8-1: 프로파일 생성 및 사용.....	69
연습 8-2: 롤 생성	71
연습 8-3: 유저 생성 및 구성	73
단원 9 의 연습.....	77
연습 9-1: Lock 충돌 해결	78
단원 10 의 연습.....	82
연습 10-1: 언두 데이터 관리	83
단원 11 의 연습.....	88
연습 11-1: 데이터베이스 감사(Audit) 구성	89
단원 12 의 연습.....	93
연습 12-1: 데이터베이스 유지 관리	94
단원 13 의 연습.....	103
연습 13-1: 성능 관리	104
단원 14 의 연습.....	114
연습 14-1: Recovery 를 위한 데이터베이스 구성	115

단원 15 의 연습.....	120
연습 15-1: 데이터베이스 백업 수행	121
단원 16 의 연습.....	128
연습 16-1: 연습 환경 준비	129
연습 16-2: 데이터 파일 손실 Recovery	130
연습 16-3: SYSTEM 테이블스페이스에서 파일 손실 Recovery.....	133
연습 16-4: 콘트롤 파일 손실 Recovery	136
단원 17 의 연습.....	140
연습 17-1: Data Pump 를 사용하여 데이터 이동	141
연습 17-2: SQL*Loader 를 사용하여 데이터 로드	147
단원 18 의 연습.....	151
연습 18-1: Critical 오류 조사	152

Oracle Internal & Oracle Academy
Use Only

단원 1의 연습

현재 상황: 이 연습에서는 오라클 구조 구성 요소에 대한 개념을 검토하고 단원에서 배운 개념에 대한 지식을 테스트하는 질문에 답합니다.

Oracle Internal & Oracle Academy
Use Only

연습 1-1: 오라클 데이터베이스 구조 탐색

빈칸에 정답을 기입하십시오.

- 1) 기본 오라클 데이터베이스 시스템은 다음 두 가지 구성 요소로 이루어집니다.

_____ 및 _____

힌트: 페이지 1-6 참조

- 2) Instance는 _____ 및 _____ 프로세스로 구성됩니다.

힌트: 페이지 1-6 참조

- 3) 세션은 _____ 프로세스와 _____ 프로세스 간의 연결입니다.

힌트: 페이지 1-8 참조

- 4) SGA의 기본 구성 요소 이름을 적으십시오.

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

힌트: 페이지 1-9 참조

연습 1-1: 오라클 데이터베이스 구조 탐색 (계속)

- 5) 오라클 데이터베이스 Instance가 포함할 수 있는 여러 백그라운드 프로세스 중에서 6개를 나열하십시오.

- _____
- _____
- _____
- _____
- _____
- _____

힌트: 페이지 1-21 참조

- 6) _____ 프로세스는 더티 버퍼를 데이터 파일에 기록합니다.

힌트: 페이지 1-23 참조

- 7) _____ 프로세스는 리두 항목을 온라인 리두 로그 파일에 기록합니다.

힌트: 페이지 1-25 참조

- 8) 오라클 데이터베이스와 연관된 기본 파일:

- _____
- _____
- _____

추가로 중요한 파일을 적으십시오.

- _____
- _____
- _____
- _____
- _____

힌트: 페이지 1-33 참조

연습 1-1: 오라클 데이터베이스 구조 탐색 (계속)

9) 오라클 데이터베이스의 논리적 저장 영역 구조:

- _____
- _____
- _____
- _____
- _____

힌트: 페이지 1-35 참조

10) _____ 프로세스는 리두 로그 파일을 아카이브 대상으로 복사합니다.

힌트: 페이지 1-31 참조

11) _____에는 서버 또는 백그라운드 프로세스에 대한 데이터 및 제어 정보가 포함됩니다.

힌트: 페이지 1-17 참조

12) 논리적 테이블스페이스 구조는 디스크의 물리적 _____ 파일과 연관됩니다.

힌트: 페이지 1-35 참조

13) LGWR가 기록하는 경우:

- _____
- _____
- _____
- _____

힌트: 페이지 1-25 참조

연습 1-1: 오라클 데이터베이스 구조 탐색 (계속)

14) 다음 문장에서 참/거짓을 표시하십시오.

- a) SGA는 데이터베이스 버퍼 캐시 및 리두 로그 버퍼를 포함합니다. _____
- b) 각 서버 프로세스에는 고유한 PGA가 포함됩니다. _____
- c) 데이터베이스 버퍼 캐시의 버퍼는 가장 최근에 사용된 리스트 및 LRU(Least Recently Used) list의 두 리스트로 구성됩니다. _____
- d) User process는 Oracle Instance에 연결되는 응용 프로그램 또는 도구를 실행합니다. _____
- e) 오라클 데이터베이스 프로세스는 서버 프로세스와 백그라운드 프로세스를 포함합니다. _____
- f) 체크포인트는 로그 파일 헤더에 기록됩니다. _____

힌트: 페이지 1-9, 1-10, 1-13, 1-20, 1-21, 1-27 참조

단원 2의 연습

현재 상황: 본 과정의 연습을 수행하는 동안 여러분이 데이터베이스 관리자(DBA) 역할을 담당한다고 가정합니다. 컴퓨터의 운영 체제(OS) 설정은 다음과 같습니다.

- 암호가 oracle인 oracle 유저
- 암호가 oracle인 root 유저

시스템 관리자가 Oracle 소프트웨어를 설치할 수 있도록 OS를 설정했습니다.

여러분은 두 가지 설치를 수행 중입니다. 첫번째 설치는 독립형 서버를 위한 Oracle Grid Infrastructure입니다. 두번째 설치는 Oracle Database 11g 소프트웨어입니다.

설치 Media는 다음 위치에 준비됩니다.

- /stage/11.2.0/clusterware/Disk1. Oracle Grid Infrastructure의 경우
- /stage/11.2.0/database/Disk1. Oracle Database 11g의 경우

별도의 지침이 없는 경우 기본 oracle OS 유저로 다음 작업을 수행하십시오.

주: 이 연습은 다음에 진행되는 모든 연습 세션에서 중요하므로 반드시 마쳐야 합니다.

연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성

이 연습에서는 독립형 서버를 위한 Oracle 그리드 Infrastructure를 설치합니다. 설치 중에는 데이터베이스의 데이터베이스 파일을 저장하는 데 사용되는 +DATA ASM 디스크 그룹을 구성합니다. 설치가 완료되면 데이터베이스의 데이터베이스 백업 및 기타 데이터베이스 파일에 사용되는 +FRA ASM 디스크 그룹을 구성합니다.

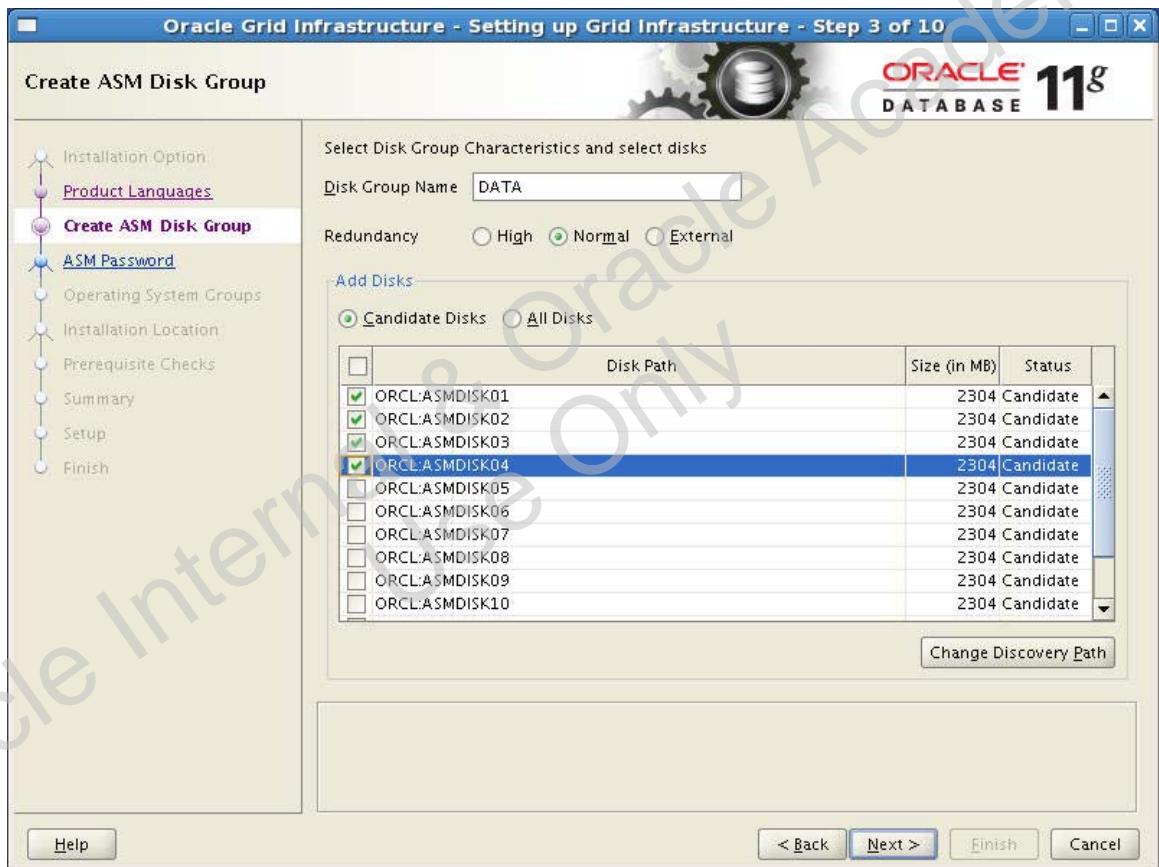
- 1) Oracle 그리드 Infrastructure용 OUI(Oracle Universal Installer)를 시작합니다.

oracle 유저로 /stage/11.2.0/clusterware/Disk1디렉토리로 이동하고 ./runInstaller를 입력합니다.

- a) 데스크톱을 마우스 오른쪽 버튼으로 누르고 Open Terminal을 눌러 터미널 window를 엽니다. 그리고 다음을 입력합니다.

```
$ cd /stage/11.2.0/clusterware/Disk1  
$ ./runInstaller
```

- 2) Select Installation Option 페이지에서 독립형 서버 옵션을 위한 Install and Configure Grid Infrastructure를 선택하고 Next를 누릅니다.
- 3) Product Languages 페이지에서 사용 가능한 모든 언어를 선택하고 Next를 누릅니다.
- 4) Create ASM Disk Group 페이지에서 Disk Group Name으로 DATA이고 Redundancy가 Normal인지 확인합니다. 처음 4개의 디스크 그룹(ORCL:ASMDISK01, ORCL:ASMDISK02, ORCL:ASMDISK03 및 ORCL:ASMDISK04)을 선택한 다음 Next를 누릅니다.



연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성 (계속)

- 5) Specify ASM Password 페이지에서 SYS 및 SYSTEM 계정에 대해 동일한 암호를 사용하도록 옵션을 선택합니다. **oracle_4U**를 암호로 입력하고 Next를 누릅니다.
- 6) 그러면 Privileged Operating System Groups 페이지가 표시됩니다. 이 설치는 독립형 서버를 위한 설치이므로 표시된 모든 관리 그룹에 대해 동일한 운영 체제 그룹(dbा)을 사용할 수 있습니다.
 - a) 3개 옵션 모두에 대해 dbा를 선택하고 Next를 눌러 계속합니다.



- b) 모든 관리자 그룹에 대해 동일 운영 체제 그룹을 지정했기 때문에 경고가 표시됩니다. 예상된 경고이므로 Yes를 눌러 계속합니다.

연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성 (계속)

- 7) Installation Location 페이지에서 Oracle Base에 대한 값이 **/u01/app/oracle**인지 확인합니다. Software Location에 **/u01/app/oracle/product/11.2.0/grid**를 입력합니다. Next를 누릅니다.



- 8) Create Inventory 페이지에서 기본값을 모두 그대로 사용하고 Next를 눌러 계속합니다.
- 9) 그러면 Perform Prerequisite Checks 페이지가 표시됩니다. OUI는 유저 환경이 이 설치의 최소 요구 사항을 충족하는지 확인합니다. 필요 조건 확인이 실패하는 많은 경우에는 OUI가 문제 해결을 위한 수정 스크립트를 생성할 수 있습니다. 본 강의에서는 모든 필요 조건이 충족되어 있으므로 아무 문제도 발생하지 않으며, OUI가 다음 페이지로 자동으로 진행합니다.
- 10) Summary 페이지에서 설정 및 정보를 검토한 다음 Finish를 누릅니다.

연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성 (계속)

11) Setup 페이지가 나타나고 설치 진행률과 수행 중인 개별 작업의 상태를 보여 줍니다. Execute Configuration 스크립트 window가 표시되면 window에 나열된 단계를 따릅니다.

a) 터미널 window를 열고 root로 로그인합니다.

```
$ su -  
Password:  
#
```

b) Execute Configuration 스크립트 window에 표시된 스크립트를 실행합니다.

i) 첫번째 스크립트는

/u01/app/oraInventory/orainstRoot.sh입니다.

```
# /u01/app/oraInventory/orainstRoot.sh  
Changing permissions of /u01/app/oraInventory.  
Adding read,write permissions for group.  
Removing read,write,execute permissions for world.  
  
Changing groupname of /u01/app/oraInventory to oinstall.  
The execution of the script is complete.  
#
```

ii) 두번째 스크립트는

/u01/app/oracle/product/11.2.0/grid/root.sh입니다.

메시지가 표시되면 Enter를 눌러 bin 디렉토리에 대해 기본값

/usr/local/bin을 사용합니다.

```
# /u01/app/oracle/product/11.2.0/grid/root.sh  
Running Oracle 11g root.sh script...  
  
The following environment variables are set as:  
ORACLE_OWNER= oracle  
ORACLE_HOME= /u01/app/oracle/product/11.2.0/grid  
  
Enter the full pathname of the local bin directory:  
[/usr/local/bin]:  
Copying dbhome to /usr/local/bin ...  
Copying oraenv to /usr/local/bin ...  
Copying coraenv to /usr/local/bin ...  
  
Creating /etc/oratab file...  
Entries will be added to the /etc/oratab file as needed by  
Database Configuration Assistant when a database is created  
Finished running generic part of root.sh script.  
Now product-specific root actions will be performed.  
2009-07-08 09:35:07: Checking for super user privileges  
2009-07-08 09:35:07: User has super user privileges  
2009-07-08 09:35:07: Parsing the host name  
Using configuration parameter file:  
/u01/app/oracle/product/11.2.0/grid/crs/install/  
crsconfig_params
```

연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성 (계속)

```
Creating trace directory
LOCAL ADD MODE
Creating OCR keys for user 'oracle', privgrp 'oinstall'..
Operation successful.
CRS-4664: Node edrsr12p1 successfully pinned.
Adding daemon to inittab
CRS-4123: Oracle High Availability Services has been started.
ohasd is starting

edrsr12p1      2009/07/08 09:35:39
/u01/app/oracle/product/11.2.0/grid/cdata/edrsr12p1/
backup_20090708_093539.OLR
Successfully configured Oracle Grid Infrastructure for a
Standalone Server
#
```

- c) Execute Configuration 스크립트 window에서 **OK**를 누릅니다. OUI가 남은 설치 작업을 계속 수행합니다.
- 12) Finish 페이지에서 **Close**를 눌러 독립형 서버를 위한 Oracle 그리드 Infrastructure 설치를 완료합니다.
- 13) 다음 단계는 +FRA 디스크 그룹을 구성하는 단계입니다. oracle로 로그인된 터미널 window에서 다음 단계를 수행합니다.
- a) oraenv 유ти리티를 사용하여 터미널 세션에 대한 환경을 설정합니다. ORACLE_SID를 입력하라는 메시지가 표시되면 **+ASM**을 입력합니다.

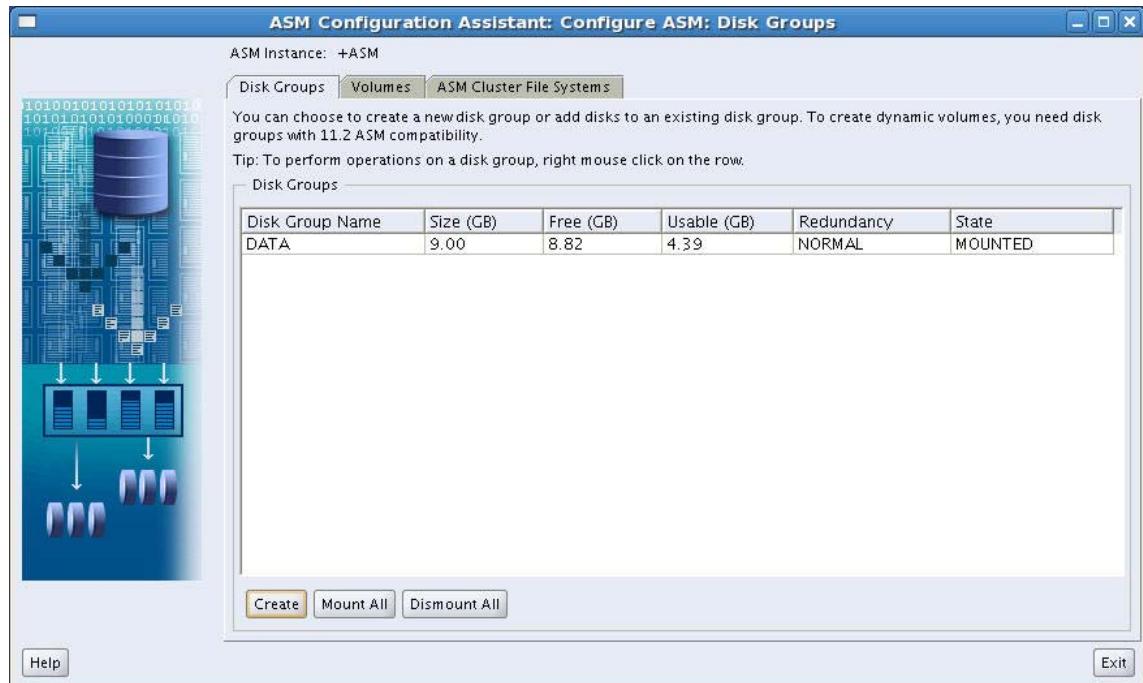
```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
```

- b) 명령행에 **asmca**를 입력하여 ASM Configuration Assistant를 시작합니다.

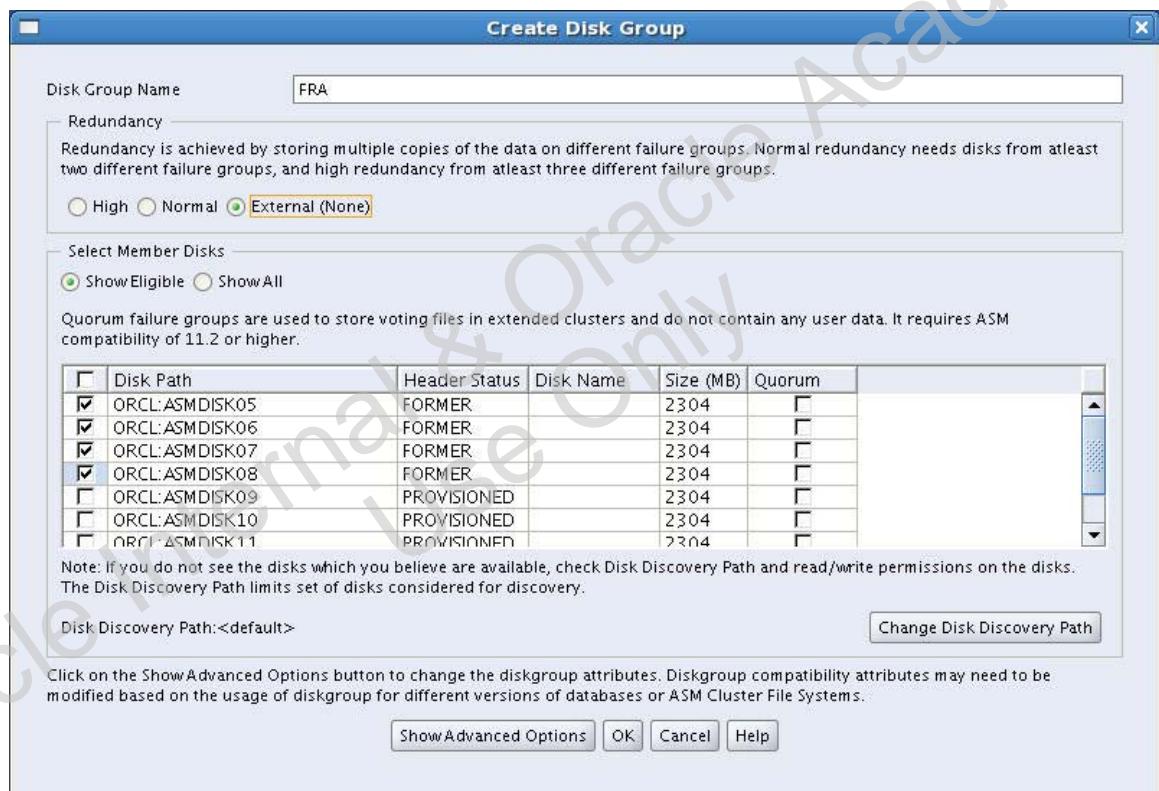
```
$ asmca
```

연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성 (계속)

- c) ASM Configuration Assistant가 열리고 +ASM Instance에 대한 현재 디스크 그룹이 표시됩니다. **Create**를 누릅니다.



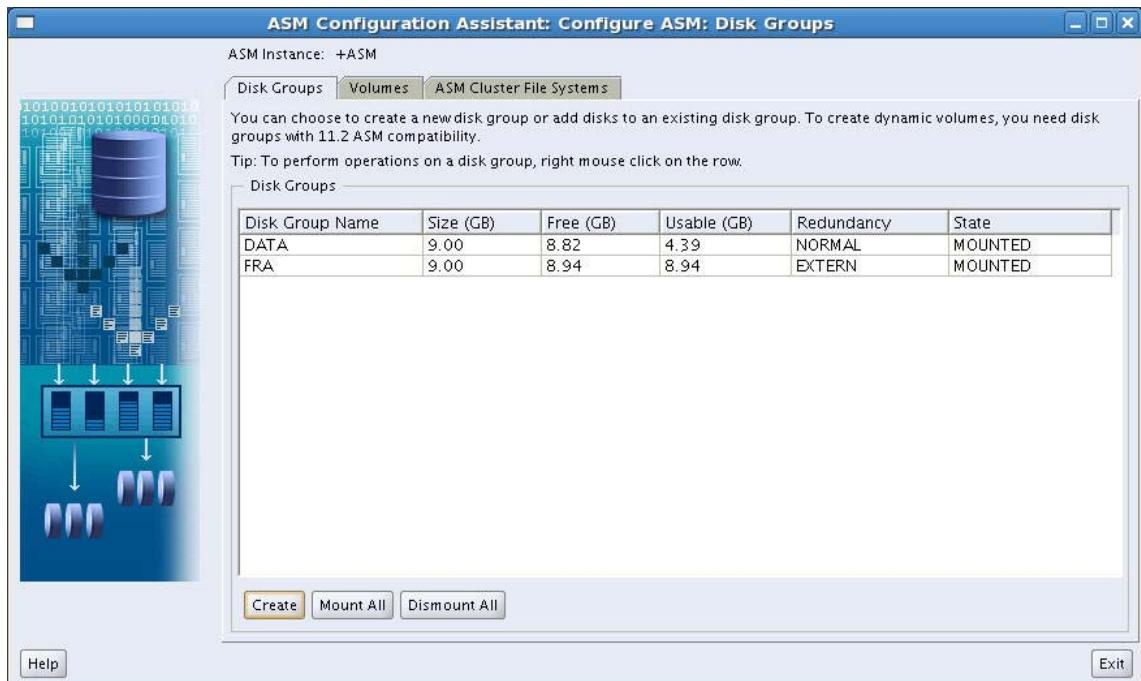
- d) Disk Group Name으로 **FRA**를 입력합니다. Redundancy에 대해 External (None)을 선택합니다. 디스크 그룹 **ORCL:ASMDISK05, ORCL:ASMDISK06, ORCL:ASMDISK07 및 ORCL:ASMDISK08**을 선택하고 **OK**를 누릅니다.



- e) DiskGroup: Creation window가 표시되면 이 window에서 **OK**를 누릅니다.

연습 2-1: Oracle 그리드 Infrastructure 설치 및 구성 (계속)

- f) 이제 +ASM Instance에 대해 두 개의 디스크 그룹(DATA 및 FRA)이 나열되어 있습니다. **Exit**를 누릅니다.



- g) 이 응용 프로그램을 종료할지 묻는 메시지가 표시되면 **Yes**를 누릅니다.

14) 열려 있는 모든 터미널 window를 닫습니다.

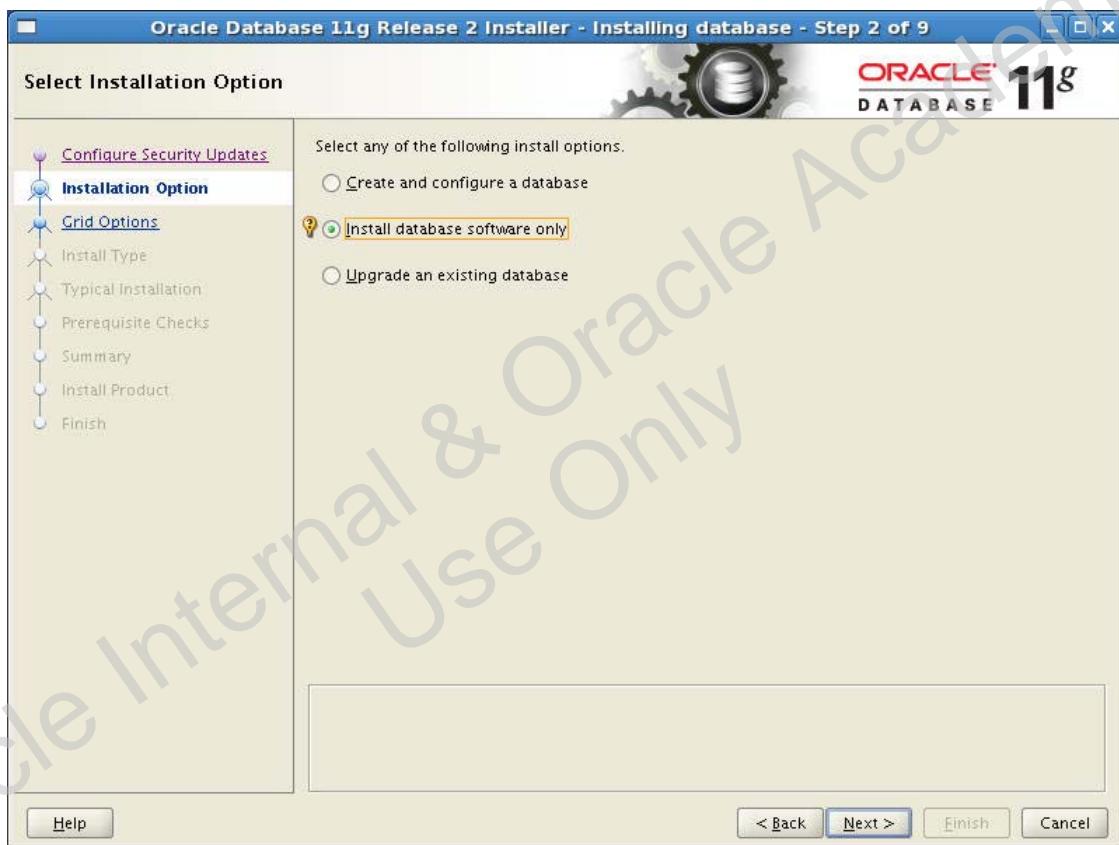
연습 2-2: Oracle Database 11g 소프트웨어 설치

다음 단계는 Oracle Database 11g 소프트웨어를 설치하는 단계입니다.

- 1) 새 터미널 window를 열고 oracle 유저로 오라클 데이터베이스 소프트웨어를 설치합니다. **/stage/11.2.0/database/Disk1** 디렉토리로 이동한 다음 **./runInstaller**를 입력하여 OUI(Oracle Universal Installer)를 시작합니다.
 - a) 데스크톱을 마우스 오른쪽 버튼으로 누르고 Open Terminal을 눌러 터미널 window를 엽니다. 그리고 다음을 입력합니다.

```
$ cd /stage/11.2.0/database/Disk1  
$ ./runInstaller
```

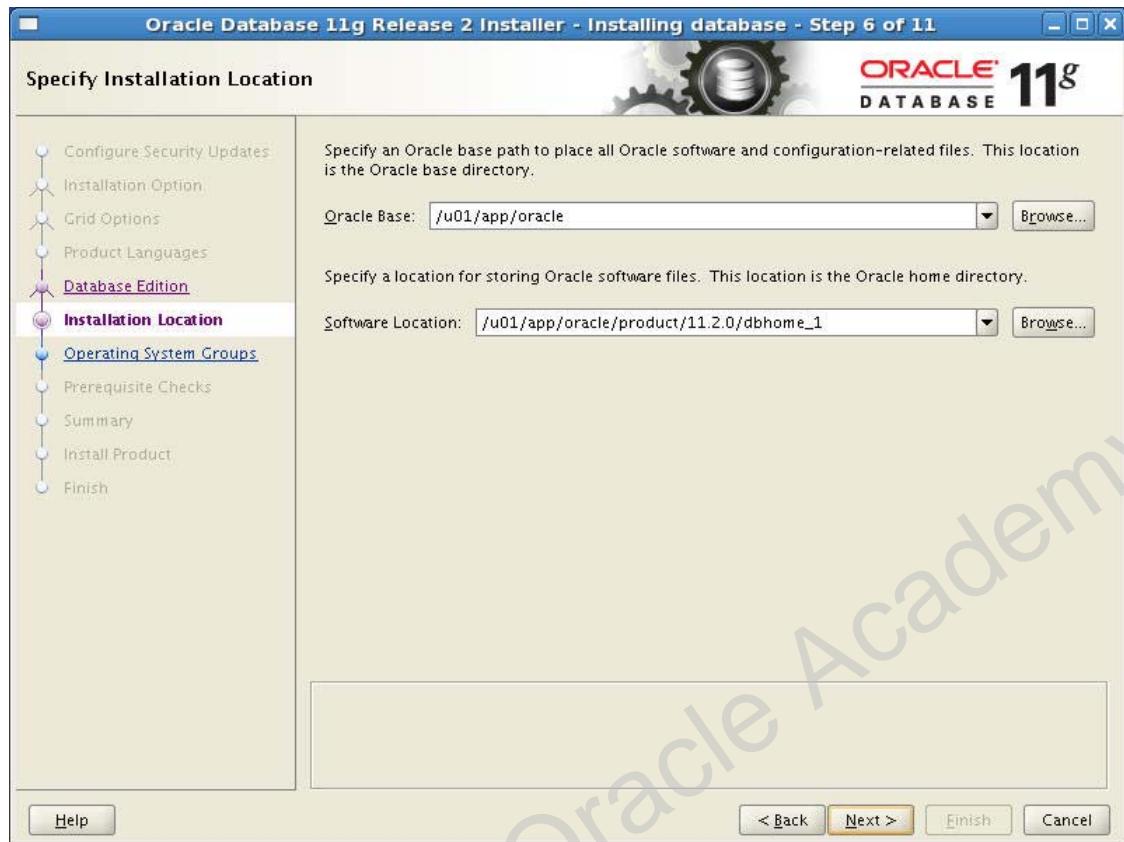
- 2) Configure Security Updates 페이지가 먼저 표시됩니다. 실제 환경에서는 전자 메일 주소와 My Oracle Support 암호를 입력하지만, 강의실은 격리된 환경이기 때문에 전자 메일 및 암호 필드를 비워 둡니다. My Oracle Support에서 보안 갱신 받기 옵션을 선택 해제하고 **Next**를 누릅니다.
- 3) Email Address Not Specified 경고 메시지가 표시되면 **Yes**를 누릅니다.
- 4) Installation Option 페이지에서 **Install database software only** 옵션을 선택하고 **Next**를 누릅니다.



- 5) Install Type 페이지에서 **Single instance database installation**이 선택되었는지 확인하고 **Next**를 누릅니다.

연습 2-2: Oracle Database 11g 소프트웨어 설치 (계속)

- 6) Product Languages 페이지에서 사용 가능한 모든 언어를 선택하고 **Next**를 누릅니다.
- 7) Select Database Edition 페이지에서 **Enterprise Edition (3.95GB)**이 선택되었는지 확인하고 **Next**를 누릅니다.
- 8) Installation Location 페이지에서 Oracle Base에 대한 값이 **/u01/app/oracle**인지 확인합니다. Software Location에 **/u01/app/oracle/product/11.2.0/dbhome_1**을 입력합니다. **Next**를 누릅니다.



- 9) Privileged Operating System Groups 페이지에서 OSDBA 및 OSOPER 그룹에 대해 모두 **dba**를 선택하고 **Next**를 누릅니다.
- 10) 그러면 OUI가 필요 조건 확인을 수행합니다. 그러면 문제가 발생하지 않고 Summary 페이지가 표시됩니다. **Finish**를 누릅니다.

연습 2-2: Oracle Database 11g 소프트웨어 설치 (계속)

11) Install Product 페이지가 표시되어 설치 진행률과 수행 중인 개별 작업의 상태를 보여 줍니다. Execute Configuration 스크립트 window가 표시되면 window에 나열된 단계를 따릅니다.

- 터미널 window를 열고 root로 로그인합니다.

```
$ su -  
Password:  
#
```

- Execute Configuration 스크립트 window에 표시된 스크립트를 실행합니다.
로컬 bin 디렉토리에 대한 기본값을 사용하고 파일을 겹쳐쓰지 않습니다.(기본 옵션을 겹쳐쓸 것이 아니므로 단순히 [Enter]를 누르면 됩니다.)

```
# /u01/app/oracle/product/11.2.0/dbhome_1/root.sh  
Running Oracle 11g root.sh script...  
  
The following environment variables are set as:  
ORACLE_OWNER= oracle  
ORACLE_HOME= /u01/app/oracle/product/11.2.0/dbhome_1  
  
Enter the full pathname of the local bin directory:  
[/usr/local/bin]:  
The file "dbhome" already exists in /usr/local/bin.  
Overwrite it? (y/n)  
[n]:  
The file "oraenv" already exists in /usr/local/bin.  
Overwrite it? (y/n)  
[n]:  
The file "coraenv" already exists in /usr/local/bin.  
Overwrite it? (y/n)  
[n]:  
  
Entries will be added to the /etc/oratab file as needed by  
Database Configuration Assistant when a database is created  
Finished running generic part of root.sh script.  
Now product-specific root actions will be performed.  
Finished product-specific root actions.  
#
```

- Execute Configuration 스크립트 window에서 OK를 누릅니다.

12) Finish 페이지에서 Close를 눌러 Oracle Database 11g 소프트웨어 설치를 완료합니다.

단원 3의 연습

현재 상황: 첫번째 오라클 데이터베이스를 생성하려고 하는데 앞으로 여러 유사한 데이터베이스가 필요할 것 같습니다. 이에 따라 orcl 데이터베이스는 물론 데이터베이스 템플리트와 데이터베이스 생성 스크립트도 생성하려고 합니다. /home/oracle/labs 디렉토리(이 과정 전체에서 가장 자주 사용하게 될 디렉토리)에서 스크립트를 찾습니다.

Oracle Internal & Oracle Academy
Use Only

연습 3-1: 오라클 데이터베이스 생성

이 연습에서는 orcl 데이터베이스를 생성합니다. DBCA(Database Configuration Assistant)를 사용하여 데이터베이스를 생성합니다.

- 1) DBCA(Database Configuration Assistant)를 시작합니다.
 - a) 데스크톱을 마우스 오른쪽 버튼으로 누르고 **Open Terminal**을 선택하여 **oracle** 유저로 터미널 window를 엽니다.
 - b) **oraenv**를 사용하여 ORACLE_HOME 환경 변수를 설정합니다. SID에 대해 **orcl**을 입력한 다음 ORACLE_HOME에 대해 **/u01/app/oracle/product/11.2.0/dbhome_1**을 입력합니다.
주: 아직 orcl 데이터베이스가 존재하지 않으므로 전체 ORACLE_HOME 경로를 입력합니다. 데이터베이스가 생성된 다음에는 SID로 orcl만 입력하면 올바른 ORACLE_HOME이 확인됩니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
ORACLE_HOME = [/home/oracle] ?
u01/app/oracle/product/11.2.0/dbhome_1
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

- c) DBCA를 시작하려면 다음을 입력합니다.

```
$ dbca
```

- 2) Welcome 페이지에서 Next를 눌러 orcl 데이터베이스 생성을 시작합니다.
- 3) Operations 페이지에서 **Create a Database**를 선택한 다음 Next를 누릅니다.
- 4) Database Templates 페이지에서 **General Purpose** 또는 **Transaction Processing** 템플릿을 선택합니다.
 - a) **Show Details**를 누르고 다음 질문에 답합니다.
 - i) **질문 1:** 몇 개의 콘트롤 파일이 생성되었습니까?

해답: 2개

주: 이 연습에서는 나중에 저장 영역 기술로 ASM을 사용하도록 선택할 때 위치가 변경됩니다.

- ii) **질문 2:** 몇 개의 리드 로그 그룹이 생성되었습니까?

해답: 3개

주: 이 연습에서는 나중에 저장 영역 기술로 ASM을 사용하도록 선택할 때 위치가 변경됩니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

iii) 질문 3: 데이터베이스 블록 크기(db_block_size)는 얼마입니까?

해답: 8KB

iv) 질문 4: Sample Schemas의 값은 무엇입니까?

해답: Sample Schemas는 False로 설정됩니다.

주: 이 연습에서는 나중에 HR 예제 스키마가 데이터베이스에 포함되도록 이 설정을 변경합니다.

v) 질문 5: Database Character Set의 템플릿 기본값은 무엇입니까?

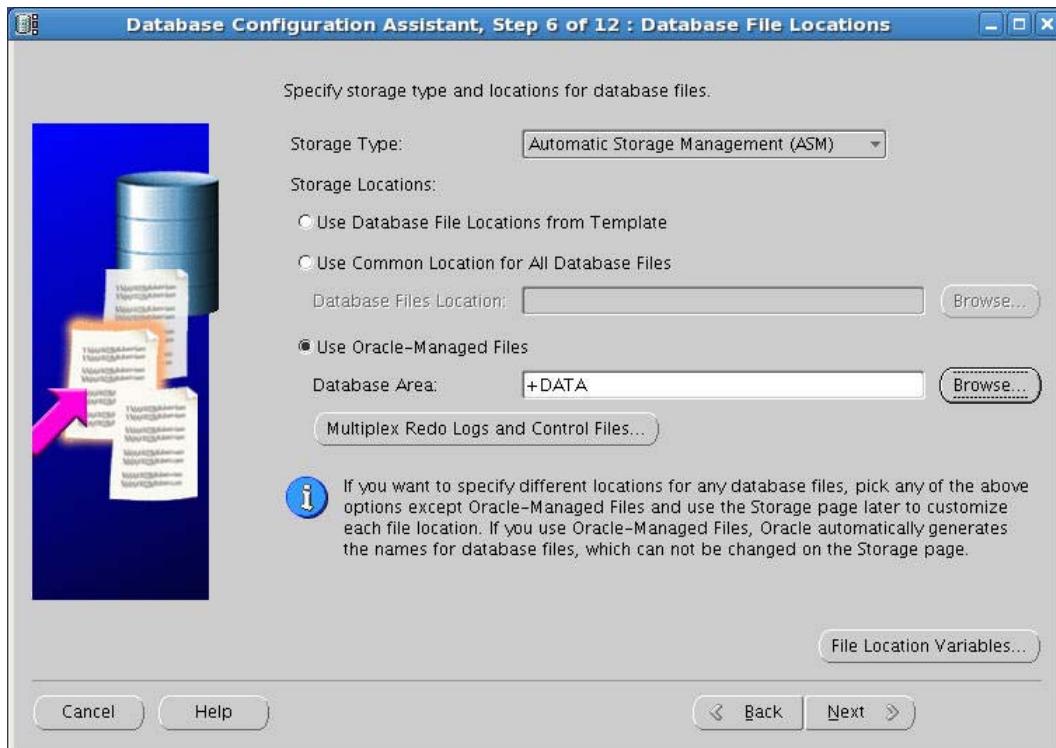
해답: WE8MSWIN1252

주: 이 연습의 뒷부분에서 유니코드 데이터베이스 Character Set를 사용하도록 이 설정을 변경하게 됩니다.

- b) Close를 눌러 Template Details window를 닫습니다.
- c) Database Templates 페이지에서 Next를 눌러 데이터베이스 생성 프로세스를 계속합니다.
- 5) Database Identification 페이지에서 Global Database Name에 **orcl.example.com**을 입력합니다. SID의 기본값은 **orcl**이라는 데이터베이스 이름으로 설정됩니다. Next를 누릅니다.
- 6) Management Options 페이지에서 다음 항목이 선택되었는지 확인합니다.
 - a) Enterprise Manager 탭에서 **Configure Enterprise Manager** 및 **Configure Database Control for local Management**가 모두 선택되었는지 확인합니다.
 - b) Automatic Maintenance Tasks 탭에서 **Enable automatic maintenance tasks**가 선택되었는지 확인합니다.
 - c) Next를 눌러 계속합니다.
- 7) Database Credentials 페이지에서 **Use the Same Password for All Accounts**를 선택하고 Password 및 Confirm Password에 **oracle_4U**를 입력합니다. 그런 다음 Next를 누릅니다.
- 8) Storage Options 페이지에서 저장 영역 유형으로 ASM을 지정하고 저장 영역 위치로 +DATA 디스크 그룹을 선택합니다.
 - a) 저장 영역 유형으로 **Automatic Storage Management (ASM)**을 선택합니다.
 - b) 저장 영역 위치는 기본적으로 Use Oracle-managed Files로 설정됩니다. Database Area에서 **Browse** 버튼을 누릅니다.
 - c) Select Disk Group window에서 **DATA** 디스크 그룹이 선택되었는지 확인하고 **OK**를 누릅니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

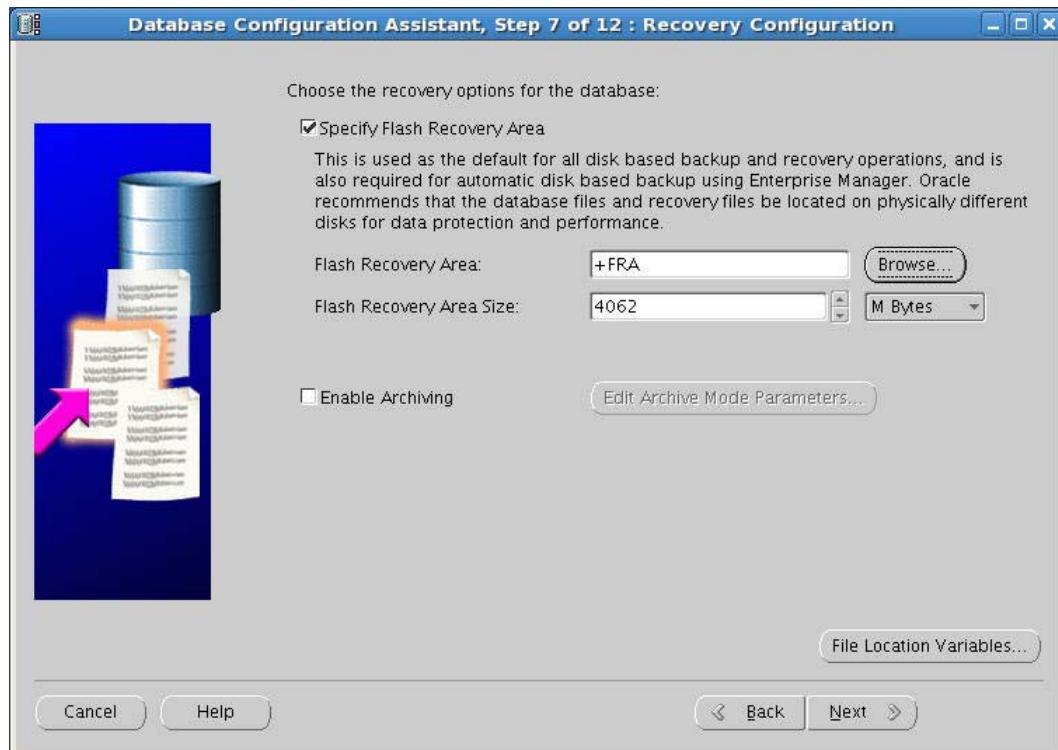
- d) Database File Locations 페이지에서 **Next**를 누릅니다.



- e) ASM Credentials window가 표시됩니다. ASM 설치 및 구성 중에 지정한 암호(강의실 환경에서 이 암호는 **oracle_4U**임)를 입력하고 **OK**를 누릅니다.
- 9) Recovery Configuration 페이지에서 Fast Recovery Area를 구성합니다.
주: Flash Recovery Area는 이제 Fast Recovery Area이지만 이 릴리스에서는 Enterprise Manager의 페이지에서 계속 Flash Recovery Area로 참조됩니다. 아직 제품 변환 과정이 진행되고 있으므로 이후 릴리스에서는 모두 변경될 예정입니다.
- Specify Flash Recovery Area를 선택합니다.
 - Flash Recovery Area에서 **Browse** 버튼을 누릅니다.
 - Select Disk Group window에서 **FRA** 디스크 그룹을 선택하고 **OK**를 누릅니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

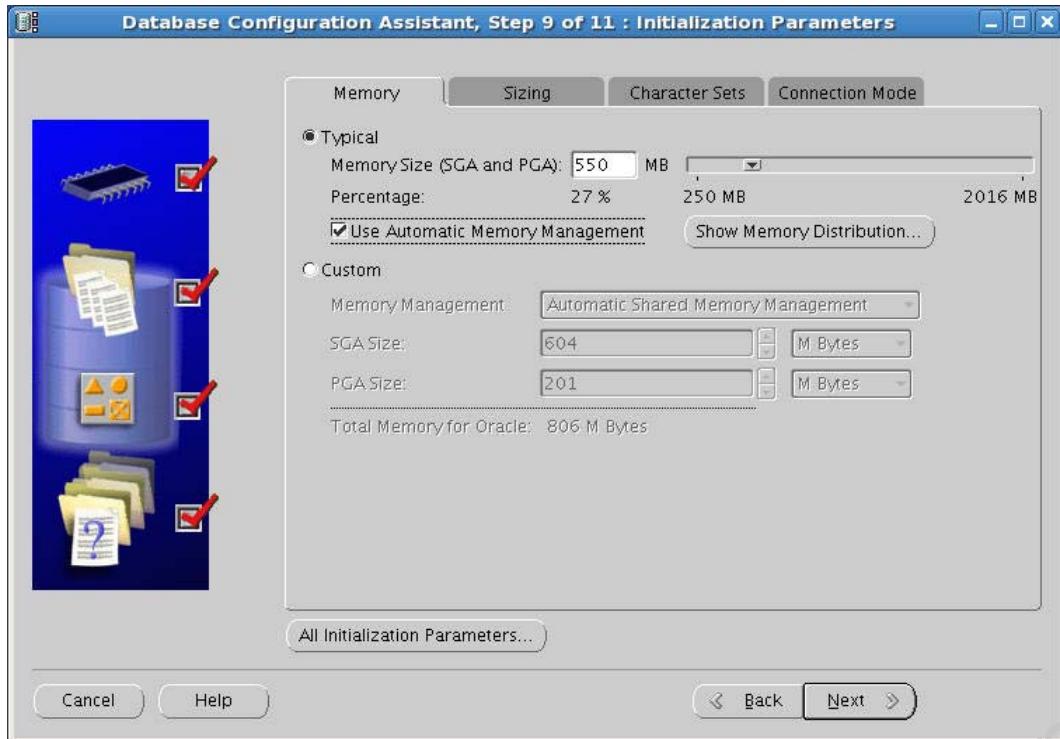
- d) Recovery Configuration 페이지에서 Next를 누릅니다.



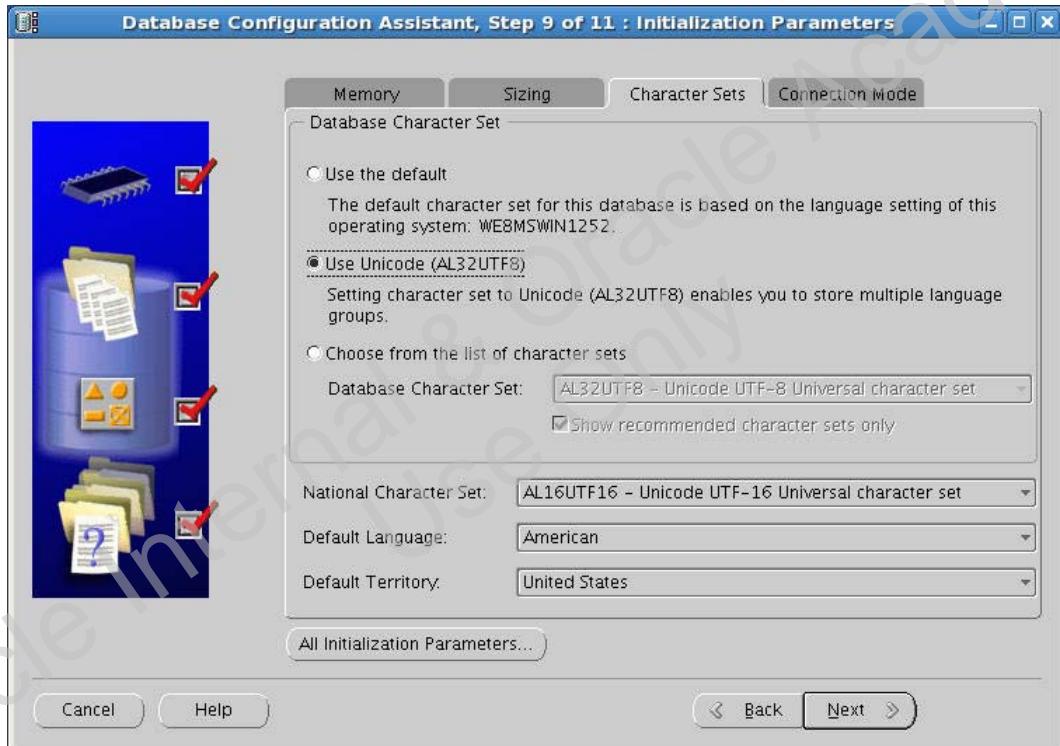
- 10) Database Content 페이지에서 **Sample Schemas**를 선택한 다음 Next를 누릅니다.
- 11) Initialization Parameters 페이지의 Memory 탭 페이지에서 **Typical**을 선택하고 Memory Size로 **550MB**를 지정합니다. **Unicode AL32UTF8**을 사용하도록 Character Set를 수정합니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

- a) Typical을 선택하고 Memory Size (SGA and PGA)에 대한 값을 550MB로 설정합니다. Memory Management 필드에 Automatic Memory Management가 선택되었는지 확인합니다.



- b) Character Sets 탭을 누르고 Use Unicode (AL32UTF8)를 선택합니다.



- c) Sizing 및 Connection Mode 탭 페이지를 검토하되 값을 변경하지는 마십시오. 그런 다음 Next를 누릅니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

12) Database Storage 페이지에서 파일 이름과 위치를 검토합니다. 그런 다음 **Next**를 누릅니다.

13) Creation Options 페이지에서 다음 항목들을 선택합니다.

- Create Database**를 선택합니다.
- Save as a Database Template** 옵션을 선택합니다. 데이터베이스 템플릿에 대한 Name에 **orcl**을 입력하고 Description에 **orcl Database Template**을 입력합니다.
- Generate Database Creation Scripts**를 선택하고 Destination Directory에 **/home/oracle/labs**를 입력합니다.
- 그런 다음 **Finish**를 누릅니다.

14) Confirmation 페이지가 나타납니다.

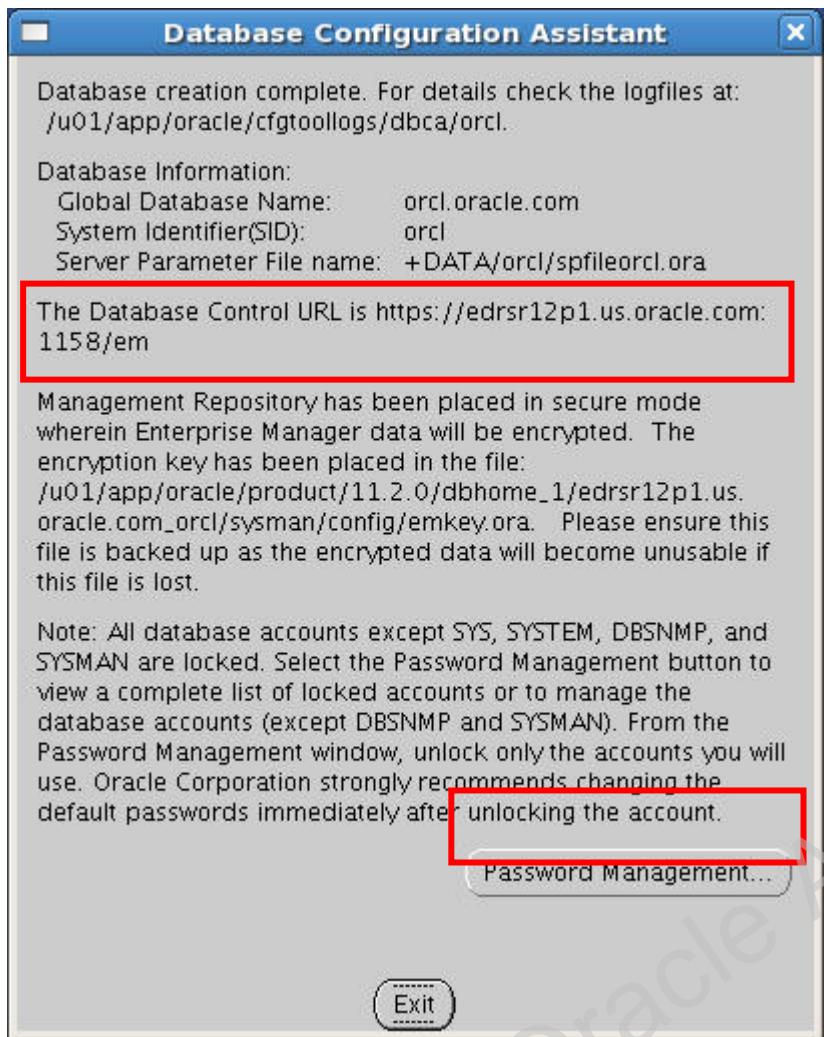
- 특히 아래 표에 있는 것들을 포함하여 옵션 및 파라미터를 검토하고 **OK**를 누릅니다.

이름	값
Sample Schemas	true
db_block_size	8KB
db_create_file_dest	+DATA
db_recovery_file_dest	+FRA
memory_target	550MB
Database Character Set	AL32UTF8

- OK**를 눌러 템플릿이 생성되었음을 승인합니다.
- OK**를 눌러 데이터베이스 스크립트 생성을 승인합니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

15) DBCA는 다양한 설치 단계의 진행 상황을 표시합니다. 데이터베이스가 생성되면 DBCA는 해당 데이터베이스에 대한 필수 정보를 표시합니다. 이 정보를 기록하십시오. Database Control URL은 다음에 진행되는 여러 연습 세션에서 사용됩니다.



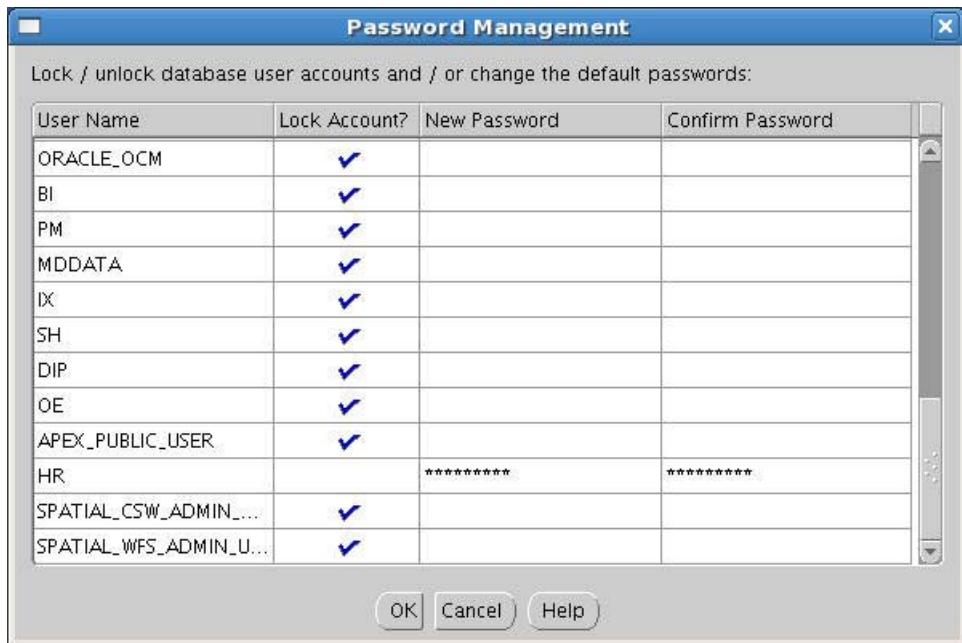
- a) **중요:** Database Control URL을 여기에 기록하십시오.

https://_____ : _____/em
이 URL은 이 과정의 남은 부분에서 여러 번 사용됩니다.

- b) **Password Management** 버튼을 누릅니다.
c) **HR** username이 나타날 때까지 Password Management 페이지를 아래 화면으로 이동합니다.

연습 3-1: 오라클 데이터베이스 생성 (계속)

- d) Lock Account?의 선택을 해제하고 New Password 및 Confirm Password에 **oracle_4U**를 입력합니다. 그런 다음 OK를 누릅니다.



- e) Exit를 눌러 DBCA를 닫습니다.

데이터베이스, 데이터베이스 템플릿 및 데이터베이스 생성 스크립트 작성 작업을 마쳤습니다.

단원 4의 연습

현재 상황: Oracle 소프트웨어를 설치하고 데이터베이스를 생성했습니다.
데이터베이스를 시작/정지할 수 있는지 확인하고 응용 프로그램 데이터를 살펴보려고 합니다.

Oracle Internal & Oracle Academy
Use Only

연습 4-1: Oracle Instance 관리

이 연습에서는 Oracle Enterprise Manager 인터페이스를 조금 더 자세히 살펴봅니다. Oracle Enterprise Manager를 사용하여 다음과 같은 작업을 수행합니다.

- Instance 파라미터 확인 및 변경
- 데이터베이스 종료
- 데이터베이스 시작

또한 SQL*Plus를 사용하여 여러 Instance 파라미터를 보고 터미널 window에서 Alert Log의 텍스트 버전을 살펴봅니다.

- 1) Enterprise Manager를 호출하고 SYS 유저로 로그인합니다. 이 데이터베이스가 사용하는 포트 번호는 무엇입니까? 연습 3에서 이 정보를 기록했습니다.

해답: 1158

- a) 데스크톱에서 **Web Browser** 아이콘을 두 번 눌러 oracle 유저로 웹 브라우저를 엽니다.

- b) 연습 3에서 기록한 URL을 입력합니다. URL은 다음과 같은 형식입니다.

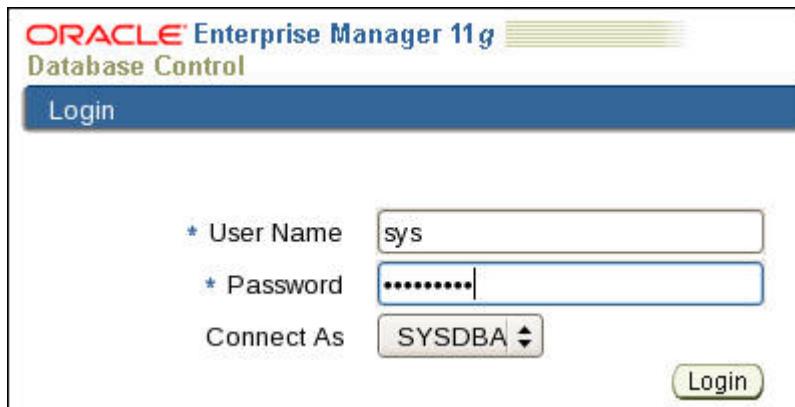
`https://hostname:portnumber/em`

주: 처음 연결하면 Secure Connection Failed 메시지(또는 유사 메시지)가 표시되고 Alert window가 표시될 수 있습니다. 이 메시지를 통과하려면 예외를 추가하고 인증서를 수락합니다.

- i) Alert window가 표시되면 **OK**를 누릅니다.
- ii) 페이지 아래에서 [Or you can add an exception?](#) 링크를 누릅니다.
- iii) 예외 추가와 관련된 경고가 표시됩니다. **Add Exception?** 버튼을 누릅니다.
- iv) Add Security Exception window에서 **Get Certificate** 버튼을 누릅니다.
- v) Certificate Status가 표시됩니다. 이 예외를 영구적으로 저장하는 옵션이 선택되었는지 확인하고 **Confirm Security Exception** 버튼을 누릅니다.

연습 4-1: Oracle Instance 관리 (계속)

- c) Oracle Enterprise Manager 로그인 화면에서 User Name으로 **sys**를 입력하고 Password로 **oracle_4U**를 입력한 다음 Connect As로 **SYSDBA**를 선택합니다. 그런 다음 **Login**을 누릅니다.



- 2) 초기화 파라미터를 확인하고 JOB_QUEUE_PROCESSES 파라미터를 15로 설정합니다. 이를 위해 실행할 SQL 문은 무엇입니까?
- a) Server > Initialization Parameters(Database Configuration 섹션에서)를 선택합니다.

The screenshot shows the 'Database Instance: orcl.oracle.com' configuration page. The 'Server' tab is active. On the left, under 'Storage', there are links for Control Files, Tablespaces, Temporary Tablespace Groups, Datafiles, Rollback Segments, and Redo Log Groups. On the right, under 'Database Configuration', there are links for Memory Advisors, Automatic Undo Management, Initialization Parameters (which is highlighted with a red box), and View Database Feature Usage.

- b) Name 필드에 **job**을 입력한 다음 Go를 누릅니다.

The screenshot shows the 'Initialization Parameters' page. The 'SPFile' tab is selected. A table lists parameters. In the 'Name' column, 'job' is selected. Other columns include 'Basic', 'Modified', 'Dynamic', and 'Category'. There are also buttons for 'All' and 'Go'. A note at the top says: 'The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFILE mode.'

- c) JOB_QUEUE_PROCESSES 초기화 파라미터가 나타나면 값을 15로 변경합니다.
- d) Show SQL을 누르고 다음에 실행할 SQL 문을 확인합니다.

The screenshot shows the 'Initialization Parameters' page again, but this time the 'Show SQL' section is visible. It contains the SQL command: 'ALTER SYSTEM SET job_queue_processes = 15 SCOPE=MEMORY'. To the right, it says 'Logged in As SYS' and there is a 'Return' button.

- e) Return을 누른 다음 Apply를 누릅니다.

연습 4-1: Oracle Instance 관리 (계속)

- 3) 질문: Dynamic 열에서 확인할 중요한 사항은 무엇입니까?
해답: "동적" 파라미터는 데이터베이스가 실행 중일 때 수정할 수 있습니다.
- 4) Enterprise Manager를 사용하여 데이터베이스 Instance를 종료합니다.
- Enterprise Manager 브라우저 세션에서 **Database** 탭을 누릅니다.
 - Shutdown** 버튼을 누릅니다.
 - Host Credentials에서 Username에 **oracle**, Password에 **oracle**을 입력합니다.
 - OK**를 누릅니다. Startup/Shutdown: Confirmation 페이지가 나타납니다.
 - Advanced Options**를 눌러 종료 모드를 확인합니다. 단, 모드는 변경하지 말고 "Immediate" 상태로 두십시오.
 - Cancel**을 눌러 이전 페이지로 돌아갑니다.
 - Yes**를 눌러 종료 작업을 확인합니다.
 - Refresh**를 누릅니다. Refresh하는 동안 오류가 나타나면 **OK**를 누르고 Refresh 작업을 계속 진행합니다. 오류가 자체적으로 해결됩니다.
 - 현재 Instance의 Status는 "Down"입니다.

The screenshot shows the 'Database Instance' section of the Oracle Enterprise Manager interface. A red arrow points to the 'Status' field, which displays 'Down'. Other details shown include Host: edrsr12p1.us.oracle.com, Port: 1521, SID: orcl, and Oracle Home: /u01/app/oracle/product/11.2.0/dbhome_1. A message in the 'Details' section states: 'There has been a user-initiated shutdown.'

- 5) SQL*Plus를 사용하여 종료된 데이터베이스에 HR 유저로 연결할 수 있는지 확인합니다.
- Linux 명령 window에서는 oraenv를 사용하여 유저 환경을 orcl 데이터베이스로 설정합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

연습 4-1: Oracle Instance 관리 (계속)

- b) 다음을 입력하여 데이터베이스에 로그인을 시도합니다.

```
$ sqlplus hr
```

- c) 암호에 **oracle_4U**를 입력합니다.
- d) "ORACLE not available" 오류 메시지가 나타납니다.
- e) [Ctrl], [D]를 눌러 username 프롬프트를 종료합니다.
- 6) Enterprise Manager를 사용하여 데이터베이스 Instance를 재시작한 다음 SYS 유저로 다시 로그인합니다.
- Enterprise Manager에서 **Startup** 버튼을 누릅니다.
 - Host Credentials 영역에서 Username 및 Password에 **oracle**을 입력합니다.
 - OK**를 누릅니다.
 - Select Startup Type 페이지가 나타납니다. 종속 리소스와 함께 데이터베이스를 시작하는 옵션이 선택되었는지 확인하고 **OK**를 누릅니다.

Select Startup Type

This database is registered with Oracle Restart. This enables you to use srvctl utility that comes with Oracle Restart to start this database. Using srvctl will attempt to start the database resource and all other resources on which this database depends (eg: listener, ASM instance etc). Alternatively you may attempt to start the database alone using sqlplus utility. Choose the way in which you want to start the database.

Select Startup Type

Start database along with dependent resources

Start database only

Cancel **OK**

- e) Startup/Shutdown: Confirmation 페이지가 나타납니다.
- f) **Advanced Options**을 눌러 시작 시 사용 가능한 모드 및 옵션을 확인하지만 모드를 변경하지는 마십시오. 시작 모드는 "Open"으로 남아 있어야 합니다.
- g) **Cancel**을 눌러 이전 페이지로 돌아갑니다.
- h) **Yes**를 눌러 시작 작업을 확인합니다.
- i) 데이터베이스가 시작되면 Startup 페이지가 표시됩니다. 로그인 페이지가 나타날 때까지 기다립니다. 이번에는 **oracle_4U** 암호 및 SYSDBA 권한을 사용하여 **SYS** 유저로 로그인할 수 있습니다.
주: 처음 연결하면 Agent Connection to Instance에 대한 상태가 Failed로 표시될 수 있습니다. 몇 분 정도 기다리면 이 상태가 사라지고 데이터베이스 홈 페이지가 표시됩니다.

연습 4-1: Oracle Instance 관리 (계속)

7) Alert Log에서 시작 중 데이터베이스가 거쳐간 단계를 확인합니다. 어떤 단계를 거쳤습니까?

- Database > Related Links > Alert Log Contents를 선택합니다. Go를 누릅니다.
- 로그를 탐색하면서 시작 중 데이터베이스 단계를 검토합니다. 시스템 동작에 따라 Alert Log는 스크린샷과 다를 수도 있습니다.

Jul 10, 2009 5:44:48 AM GMT+07:00	NOTIFICATION	16	admin_ddl opiexe:2994:4222364190	ALTER DATABASE OPEN
Jul 10, 2009 5:44:48 AM GMT+07:00	NOTIFICATION	16	admin_ddl opiexe:3065:2802784106	Completed: ALTER DATABASE MOUNT

- 시작 중 데이터베이스가 거쳐간 모드는 MOUNT 및 OPEN입니다.
- Alert Log의 텍스트 버전을 찾아 확인합니다.
SQL*Plus를 사용하여 **system** 유저(암호 **oracle_4U**)로 데이터베이스에 연결하고 **V\$DIAG_INFO** 뷰를 query합니다. XML 태그가 없는 텍스트 전용 Alert Log를 보려면 다음 단계를 수행하십시오.
 - V\$DIAG_INFO query 결과에서 Diag Trace 항목에 해당하는 경로를 기록해둡니다.

```
SQL> select * from V$DIAG_INFO;  
  
INST_ID NAME  
-----  
-----  
VALUE  
-----  
...  
    1 Diag Trace  
/u01/app/oracle/diag/rdbms/orcl/orcl/trace  
...
```

- SQL*Plus를 종료하고 해당 경로로 디렉토리를 변경합니다.
 - \$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
iii) 텍스트 편집기를 사용하여 **alert_orcl.log** 파일을 엽니다.
주: 이 파일은 다른 데이터베이스에서 alert_<sid>.log로 이름이 지정됩니다. 여기서 <sid>는 Instance 이름입니다.
 - 앞의 연습에서 수행한 종료 및 시작에 대한 항목을 찾아봅니다.
- 8) SQL*Plus를 사용하여 sysdba로 데이터베이스에 연결합니다.
주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus / as sysdba
```

연습 4-1: Oracle Instance 관리 (계속)

- 9) **SHOW PARAMETER** 명령을 사용하여 **SGA_MAX_SIZE**, **DB_CACHE_SIZE** 및 **SHARED_POOL_SIZE**에 대한 설정을 확인합니다.

```
SQL> show parameter sga_max_size
NAME                                     TYPE        VALUE
-----
sga_max_size                           big integer 552M
SQL> show parameter db_cache_size
NAME                                     TYPE        VALUE
-----
db_cache_size                           big integer 0
SQL> show parameter shared_pool_size
NAME                                     TYPE        VALUE
-----
shared_pool_size                        big integer 0
SQL>
```

- 10) **JOB_QUEUE_PROCESSES**의 값을 확인합니다.

```
SQL> show parameter job_queue_processes
NAME                                     TYPE        VALUE
-----
job_queue_processes                      integer    1000
SQL>
```

주: 이 연습의 초반부에서는 **JOB_QUEUE_PROCESSES** 파라미터를 15로 변경했지만 이러한 변경은 실행 중인 Instance만을 위한 것이었습니다. 이러한 변경 사항은 SPFILE에 저장되지 않았습니다. 따라서 다음에 종료 및 시작을 수행하면 값이 SPFILE에 있는 값으로 되돌려집니다.

연습 4-2: Oracle Restart 및 데이터베이스 테스트

이 연습에서는 데이터베이스가 중단되도록 만들고 Oracle Restart가 데이터베이스를 재시작하는지 감시하여 Oracle Restart의 기능을 테스트합니다.

- 1) Enterprise Manager를 사용하여 orcl 데이터베이스 Instance가 현재 Oracle Restart에 의해 관리되는지를 확인합니다.
 - a) Database 탭을 눌러 Home 페이지로 돌아갑니다.
 - b) Home 페이지에서 **High Availability** 섹션을 보고 Oracle Restart가 데이터베이스 Instance에 대해 **Enabled**되었는지 확인합니다.
- 2) Oracle Restart가 작동하는지 확인하려면 orcl 데이터베이스 Instance의 LGWR 프로세스를 종료합니다. 어떤 결과가 발생합니까?
 - a) orcl 데이터베이스 Instance에 대한 환경 변수를 설정합니다.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
Enter
```

- b) **ps -ef | grep ora_lgwr_orcl**을 입력하여 데이터베이스에 대한 LGWR 프로세스의 프로세스 ID를 찾습니다.

```
$ ps -ef | grep ora_lgwr_orcl
oracle  10478      1  0 10:54 ?        00:00:00 ora_lgwr_orcl
oracle  11371  24865  0 11:00 pts/3    00:00:00 grep
ora_lgwr_orcl
```

- c) **kill -9** 명령 및 이전 단계에서 확인한 프로세스 ID를 사용하여 LGWR을 종료합니다. 이렇게 하면 Instance가 종료됩니다.

```
$ kill -9 10478
```

연습 4-2: Oracle Restart 및 데이터베이스 테스트 (계속)

- d) **ps -ef | grep ora_lgwr_orcl**을 다시 입력하여 LGWR 프로세스가 재시작되는지 확인합니다. LGWR가 다시 시작된 것으로 표시될 때까지 이를 반복합니다. 이제 ora_lgwr_orcl 프로세스는 **kill -9** 명령을 실행할 때 사용한 프로세스 ID와는 다른 프로세스 ID를 갖습니다.

```
$ ps -ef | grep ora_lgwr_orcl
oracle  11849 11687  0 11:06 pts/3      00:00:00 grep
ora_lgwr_orcl
$ ps -ef | grep ora_lgwr_orcl
oracle  11855 11687  0 11:06 pts/3      00:00:00 grep
ora_lgwr_orcl
$ ps -ef | grep ora_lgwr_orcl
oracle  11946      1  0 11:06 ?
ora_lgwr_orcl
oracle  12034 11687  0 11:07 pts/3      00:00:00 grep
ora_lgwr_orcl
```

- 3) sysdba로 SQL*Plus를 사용하여 데이터베이스에 연결하고 데이터베이스가 성공적으로 재시작되었는지 확인합니다. v\$instance를 query하여 데이터베이스 상태를 확인합니다.
주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on
Tue Aug 18 11:16:40 2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> select status from v$instance;

STATUS
-----
OPEN

SQL>
```

단원 5의 연습

현재 상황: 이 연습에서는 ASM Fast Mirror Resync 기능의 이점을 살펴봅니다. 또한 ASM 명령행 유틸리티인 ASMCMD의 기능을 살펴봅니다.

Oracle Internal & Oracle Academy
Use Only

연습 5-1: ASM Fast Mirror Resync

이 연습에서는 비데이터 손실 문제(non-data-loss issue) 발생 후 오프라인 상태인 디스크를 추가하는 데 걸리는 시간을 비교합니다. 한 번은 ASM Fast Mirror Resync를 사용하지 않고 한 번은 사용하여 동일한 작업을 비교합니다.

- 1) 기존 ASM 디스크 그룹의 호환성 값을 확인합니다. 어떤 결과가 발생합니까?

- a) +ASM instance에 대해 적합한 환경 변수를 설정합니다. 다음 스크린샷의 query를 실행하여 ASM과 데이터베이스 모두에 대해 COMPATIBLE 값을 확인합니다. 데이터베이스 호환성 값이 10.1로 설정된 것을 확인할 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
$ sqlplus / as sysasm

SQL> select name, compatibility, database_compatibility from
v$asm_diskgroup;

NAME          COMPATIBILITY      DATABASE_COMPATIBILITY
-----          -----          -----
DATA           11.2.0.0.0          10.1.0.0.0
FRA            11.2.0.0.0          10.1.0.0.0
```

- 2) Enterprise Manager를 사용하여 **DATA** 디스크 그룹 페이지로 이동합니다.

- a) 이전 실습에서 제공된 URL을 사용하여 SYS 유저로 Enterprise Manager Database Control에 연결합니다. 예:

<https://edrsr12p1.us.oracle.com:1158/em/>

- b) 기본 Database 페이지에서 +ASM 링크를 누릅니다.
- c) ASM 홈 페이지에서 **Disk Groups** 탭을 누릅니다.
- d) Automatic Storage Management Login 페이지에서 Username 필드에 **sys**, Password 필드에 **oracle_4U**를 입력하고 Connect As drop-down list에서 **SYSASM**을 선택합니다. Save as Preferred Credentials 체크 박스를 선택합니다. 그런 다음 **Login**을 누릅니다.
- e) Disk Groups 페이지의 테이블에서 **DATA** 링크를 누릅니다.

연습 5-1: ASM Fast Mirror Resync (계속)

- 3) Enterprise Manager Database Control을 사용하여 **DATA** 디스크 그룹의 데이터베이스 호환성 속성을 변경합니다. **11.2.0.0.0**으로 설정합니다.
 - a) Disk Group: DATA General 하위 페이지의 Advanced Attributes 섹션에서 **Edit**를 누릅니다.
 - b) Edit Advanced Attributes for Disk Group: DATA 페이지에서 Database Compatibility에 **11.2.0.0.0**을 입력합니다. 그런 다음 **OK**를 누릅니다.
 - c) **Disk Group: DATA General** 하위 페이지로 돌아갑니다. Database Compatibility 필드가 갱신되었는지 확인합니다. 또한 디스크의 사용량이 40% 정도로 표시되어야 합니다.
 - d) Database Compatibility를 다시 버전 **10.1.0.0.0**으로 변경합니다. 어떤 결과가 발생합니까?
- 4) SQL*Plus를 사용하여 이전 갱신이 올바르게 수행되었는지 확인합니다.
 - a) oracle 유저로 연결된 터미널 window에서 SQL*Plus를 실행하고 **V\$ASM_DISKGROUP**을 확인합니다.

```
$ . 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> select name,compatibility,database_compatibility from
v$asm_diskgroup;

NAME          COMPATIBILITY      DATABASE_COMPATIBILITY
-----          -----          -----
DATA           11.2.0.0.0          11.2.0.0.0
FRA            11.2.0.0.0          10.1.0.0.0
```

- b) 작업이 끝나면 SQL*Plus를 종료합니다.

연습 5-1: ASM Fast Mirror Resync (계속)

- 5) labs 디렉토리에서 **lab_05_01_05.sh** 스크립트를 실행하여 이 연습에 대한 환경을 설정합니다. 이 스크립트는 50MB 파일을 사용하여 DATA 디스크 그룹에 TBSJMW라는 새 테이블스페이스를 생성합니다. 그런 다음 이 새 테이블스페이스에 상주하는 SYSTEM.JMW라는 새 테이블이 생성됩니다. 그런 다음 이 스크립트는 새로 생성된 테이블에 몇 개의 행을 삽입합니다.

```
$ cd ~/labs
$ ./lab_05_01_05.sh
ORACLE_SID = [oracle] ? The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle

SQL*Plus: Release 11.2.0.1.0 Production on
Wed Jul 29 04:43:05 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> SQL> SQL> SQL>
Tablespace created.

SQL> SQL> drop table jmw purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL>
Table created.

SQL> SQL>
1 row created.

SQL> SQL>
Commit complete.

SQL> SQL> 2   3   4   5   6   7
PL/SQL procedure successfully completed.

SQL> 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
$
```

연습 5-1: ASM Fast Mirror Resync (계속)

- 6) DATA 디스크 그룹에 속한 두번째 디스크를 오프라인 상태로 만들고 Disk Repair Time 속성이 0으로 설정되었는지 확인합니다.
- Disk Group: DATA General 페이지로 돌아갑니다. 두번째 디스크(**ASMDISK02**)를 선택하고 Offline을 누릅니다.
 - Confirmation 페이지에서 Disk Repair Time을 기본값(3.6시간)에서 **0.0**으로 변경하고 Show SQL을 누릅니다.

```
ALTER DISKGROUP DATA OFFLINE DISK ASMDISK02 DROP AFTER 0.0 h
```

- Return을 누릅니다.
 - Confirmation 페이지로 돌아갑니다. Yes를 누릅니다.
- 7) 어떤 결과가 발생합니까?
- Disk Group: DATA General 페이지로 돌아갑니다. ASMDISK02가 오프라인이 된 것을 확인할 수 있습니다. 오프라인 상태의 디스크가 더 이상 나타나지 않을 때까지 브라우저 페이지를 Refresh합니다. 이름은 아래와 비슷한 이름으로 바뀝니다.
_DROPPED_0000_DATA
Pending Operations는 디스크가 삭제 중이므로 1로 표시됩니다. 리밸런스 작업의 진행률을 보려면 1을 누릅니다.
 - Disk Group: DATA General 페이지로 돌아갑니다. 이제 남은 디스크 3개 모두 사용량이 54%정도로 표시됩니다. 이렇게 하여 손실된 mirrored extent가 정상적인 디스크를 통해 리밸런싱되었습니다.

- 8) **SYSTEM.JMW** 테이블에서 일부 행을 수정합니다(499개 행 삭제). 제대로 작동합니까?

- JMW 테이블을 수정할 수는 있습니다.

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus system
Enter password: oracle_4U

SQL> delete from system.jmw where rownum<500;

499 rows deleted.

SQL> commit;
```

연습 5-1: ASM Fast Mirror Resync (계속)

- 9) 삭제된 ASM 디스크를 DATA 디스크 그룹에 다시 추가합니다.
- 이제 삭제된 디스크를 정리해야 다시 추가할 수 있습니다. 이렇게 하려면 root 유저여야 합니다.

```
# oracleasm listdisks  
# oracleasm deletedisk ASMDISK02  
# oracleasm createdisk ASMDISK02 /dev/xvdc
```

- Disk Group: DATA General 페이지로 돌아갑니다. Add를 누릅니다.
- Add Disks 페이지의 Candidate Member Disks 테이블에서 ORCL:ASMDISK02를 선택합니다. REBALANCE POWER를 11로 설정합니다.
- Show SQL을 누릅니다.

```
ALTER DISKGROUP DATA ADD DISK 'ORCL:ASMDISK02' SIZE 2304 M  
REBALANCE POWER 11
```

- Return을 누릅니다.
- Add Disks 페이지에서 OK를 누릅니다.

10) 어떤 결과가 발생합니까?

- Disk Group: DATA General 페이지로 돌아갑니다. Pending Operations 1 링크를 눌러 리밸런싱 작업을 모니터합니다.
- 리밸런스 작업이 잠시 진행되는 것을 확인할 수 있습니다.
- 리밸런스가 완료되도록 둡니다. 이 작업은 몇 분 정도 소요될 수 있습니다.

11) DATA 디스크 그룹에 속하는 두번째 디스크를 오프라인으로 전환하고 Disk Repair Time 속성이 기본값(3.6 시간)으로 설정되어 있는지 확인합니다. SYSTEM.JMW 테이블을 다시 수정합니다(다른 499개 행 일괄 삭제). 어떤 결과가 발생합니까?

- Disk Group: DATA General 페이지로 돌아갑니다. 두번째 디스크(ASMDISK02)를 선택하고 Offline을 누릅니다.
- Confirmation 페이지의 Disk Repair Time 필드에서 기본값 3.6 Hours를 그대로 두고 Yes를 누릅니다.
- Disk Group: DATA General 페이지로 돌아갑니다. ASMDISK02가 비어 있는 것을 확인할 수 있습니다. 브라우저 페이지를 refresh해도 리밸런스가 발생하지 않습니다.

연습 5-1: ASM Fast Mirror Resync (계속)

- d) **SYSTEM.JMW** 테이블을 수정할 수는 있습니다.

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus system
Enter password: oracle_4U

SQL> delete from system.jmw where rownum<500;

SQL> commit;
```

12) 이제 어떤 방법으로 오프라인 상태의 디스크를 DATA 디스크 그룹에 다시 추가할 수 있습니까? 삭제된 디스크를 정리할 필요가 없습니다.

- Disk Group: DATA General** 페이지로 돌아갑니다. 오프라인 디스크를 선택하고 **Online**을 누릅니다.
- Confirmation 페이지에서 **Yes**를 누릅니다.
- Disk Group: DATA General** 페이지로 돌아갑니다. 리밸런스 작업을 수행할 필요 없이 디스크가 다시 원래 레벨(약 41% 채워짐)로 돌아간 것을 확인할 수 있습니다. 즉시 디스크가 다시 추가됩니다.

연습 5-2: ASMCMD 사용

이 연습에서는 ASMCMD 명령을 사용하여 디스크 그룹을 관리합니다.

- ASMCMD를 시작하고 **+DATA** 디스크 그룹의 내용을 봅니다. **DATAFILE** 디렉토리의 리스트를 얻습니다.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ asmcmd
ASMCMD> ls +DATA/*
+DATA/ASM/:
ASMPARAMETERFILE/
+DATA/ORCL/:
CONTROLFILE/
DATAFILE/
ONLINELOG/
PARAMETERFILE/
TEMPFILE/
Spfileorcl.ora

ASMCMD> ls +DATA/ORCL/DATAFILE
EXAMPLE.260.630800437
SYSAUX.257.628766309
SYSTEM.256.628766309
TBSJMW.269.628767357
UNDOTBS1.258.628766309
USERS.259.628766309
```

- ASMCMD를 사용하여 **help** 명령으로 허용되는 모든 명령의 리스트를 생성합니다.

```
ASMCMD> help
```

- DATA** 디스크 그룹에서 **ORCL** 데이터베이스의 **CONTROLFILE** 디렉토리로 이동하고 ASMCMD를 사용하여 현재 콘트롤 파일을 **/tmp** 디렉토리에 복사합니다. 구문 지침을 보려면 **help cp** 명령을 사용합니다.

```
ASMCMD> cd +DATA/ORCL/CONTROLFILE
ASMCMD> ls
Current.260.692183799
ASMCMD> help cp
ASMCMD> cp Current.260.692183799 /tmp
copying +DATA/ORCL/CONTROLFILE/Current.260.692183799 ->
/tmp/Current.260.692183799
```

연습 5-2: ASMCMD 사용 (계속)

- 4) 현재 콘트롤 파일을 제거해봅니다. 구문 지침을 보려면 **help rm** 명령을 사용합니다.

주: ORCL 데이터베이스 Instance가 현재 실행 중이어야 하며 **DATA** 디스크 그룹이 마운트되어 있어야 합니다.

```
ASMCMD> help rm
ASMCMD> rm Current.260.692183799
ORA-15032: not all alterations performed
ORA-15028: ASM file
'+DATA/ORCL/CONTROLFILE/Current.260.692183799' not dropped;
currently being accessed (DBD ERROR: OCIStmtExecute)
```

- 5) **lsdg** 명령에 대한 구문을 확인하고 모든 디스크 그룹에 대한 리스트를 생성합니다.

```
ASMCMD> help lsdg
ASMCMD> lsdg
State      Type      Rebal    Sector   Block       AU  Total_MB
Free_MB   Req_mir_free_MB  Usable_file_MB Offline_disks
Voting_files  Name
MOUNTED    NORMAL    N          512     4096  1048576    13824
10269           600                  4834          0
N  DATA/
MOUNTED    EXTERN    N          512     4096  1048576    9216
8982           0                  8982          0
N  FRA/
```

- 6) **mkdg** 명령에 대한 구문을 확인하고 **ORCL:ASMDISK11** 및 **ORCL:ASMDISK12**의 두 디스크를 사용하여 **external** redundancy 유형인 **DATA2**라는 새 디스크 그룹을 생성합니다. 디스크 그룹이 성공적으로 생성되었는지 확인합니다.

```
ASMCMD> help mkdg
ASMCMD> mkdg <dg name="DATA2" redundancy="external"> <dsk
string="ORCL:ASMDISK11" /> <dsk string="ORCL:ASMDISK12" />
</dg>
ASMCMD> lsdg
State      Type      Rebal    Sector   Block       AU  Total_MB
Free_MB   Req_mir_free_MB  Usable_file_MB Offline_disks
Voting_files  Name
MOUNTED    NORMAL    N          512     4096  1048576    13824
10269           600                  4834          0
N  DATA/
MOUNTED    EXTERN    N          512     4096  1048576    4608
4556           0                  4556          0
N  DATA2/
MOUNTED    EXTERN    N          512     4096  1048576    9216
8982           0                  8982          0
N  FRA/
```

연습 5-2: ASMCMD 사용 (계속)

- 7) **dropdg** 명령에 대한 구문을 확인하고 마지막 단계에서 생성한 **DATA2** 디스크 그룹을 삭제합니다. 결과를 확인하십시오.

```
ASMCMD> help dropdg
ASMCMD> dropdg DATA2
ASMCMD> lsdg
State      Type    Rebal   Sector   Block       AU  Total_MB
Free_MB   Req_mir_free_MB  Usable_file_MB Offline_disks
Voting_files  Name
MOUNTED   NORMAL   N          512     4096  1048576    13824
10269           600             4834          0
N  DATA/
MOUNTED   EXTERN   N          512     4096  1048576    9216
8982            0             8982          0
N  FRA/
```

단원 6의 연습

현재 상황: 이 연습에서는 동료의 시스템 중 하나에서 시스템과 데이터베이스 사이의 연결을 구성합니다. 또한 추가 리스너를 구성하고 테스트합니다. 이 연습은 완전히 교육만을 위한 것이며, 이후 연습은 이 연습의 성공 여부에 관계없이 진행됩니다.

Oracle Internal & Oracle Academy
Use Only

연습 6-1: 다른 데이터베이스를 액세스하기 위한 Oracle 네트워크 구성

파트너의 orcl 데이터베이스에 연결할 수 있도록 네트워크 환경을 구성합니다. 로컬 이름 지정을 사용하여 파트너의 orcl 데이터베이스에 매핑되는 testorcl이라는 새 네트워크 서비스 이름을 생성합니다. testorcl 서비스 이름을 사용해서 파트너의 데이터베이스에 연결을 시도하여 네트워크 변경 사항을 테스트합니다.

- 1) **tnsnames.ora** 파일의 복사본을 만듭니다. 이 파일은 데이터베이스 **\$ORACLE_HOME/network/admin** 디렉토리에 있습니다.

- a) 터미널 window에서 oraenv를 사용하여 데이터베이스 홈에 맞게 환경을 설정합니다.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

- b) **cd \$ORACLE_HOME/network/admin**을 입력하여 **/u01/app/oracle/product/11.1.0/db_1/network/admin** 디렉토리로 이동합니다.
- c) **cp tnsnames.ora tnsnames.old**를 입력하여 **tnsnames.ora** 파일의 복사본을 생성합니다.
- d) 디렉토리에 있는 복사본 및 해당 복사본의 권한을 보려면 **ls -l**을 입력합니다.

- 2) Net Services Administration 페이지로 이동합니다. Database home 페이지에서 Listener 링크를 눌러 시작합니다.

- a) orcl 데이터베이스에 대해 **SYSDBA** 룰의 **SYS** 유저로 Enterprise Manager를 호출합니다.
- b) Database Instance – Home 페이지의 General 영역에서 Listener 링크를 누릅니다.
- c) Related Links 영역에서 Net Services Administration을 누릅니다.

연습 6-1: 다른 데이터베이스를 액세스하기 위한 Oracle 네트워크 구성 (계속)

- 3) 다른 데이터베이스에 연결할 수 있도록 로컬 Names Resolution 파일을 수정합니다. 파트너의 orcl 데이터베이스에 대한 연결 이름을 **testorcl**로 지정합니다.
- Net Services Administration 페이지의 Administer drop-down list에서 **Local Naming**을 선택한 다음 **Go**를 누릅니다.
 - Netservices Administration: Host Login 페이지가 나타납니다. 이전에 호스트 로그인에 대한 선호 인증서로 Username oracle과 Password oracle을 저장한 경우 화면에 해당 항목이 나타납니다. 그렇지 않은 경우 Username과 Password에 **oracle**을 입력하고 **Save as Preferred Credential** 체크 박스를 선택한 다음 **Login**을 누릅니다.
 - Local Naming 페이지에서 **Create**를 누르고 새 네트워크 서비스 이름을 입력합니다.
 - Net Service Name에 **testorcl**을 입력합니다.
 - Use Service Name을 선택하고 Service Name에 **orcl.example.com**을 입력합니다.
주: Use SID 옵션을 선택하여 SID를 입력하도록 선택할 수도 있습니다. 이 경우 orcl을 입력해야 합니다.
 - Database Default를 선택합니다.

Create Net Service Name

General Advanced Cancel OK

* Net Service Name

Database Information
To identify the database or service, you must provide either its service name (recommended) or the Oracle System Identifier (SID). The service name is normally its global database name, a name comprising the database name and domain name.

Use Service Name
Service Name

Use SID
SID

Choose if you want a shared or dedicated server database connection.

Database Default
Requests will be served by whatever database default is.

Dedicated Server
Requests will be served by dedicated server.

Shared Server
Request will be served by shared server.

연습 6-1: 다른 데이터베이스를 액세스하기 위한 Oracle 네트워크 구성 (계속)

- g) Addresses 영역에서 Add를 누릅니다.
- h) Add Address 페이지에서 다음 값을 지정합니다.

옵션	값
Protocol	TCP/IP
Port	1521
Host	<파트너의 호스트 이름 또는 IP 주소. 예: edrsr25p1.us.oracle.com or 139.185.35.125 >



- i) OK를 눌러 Create Net Service Name 속성 페이지로 돌아갑니다.
- j) OK를 누릅니다.
"Net Service "testorcl" created successfully."라는 Creation Message가 나타납니다.

연습 6-1: 다른 데이터베이스를 액세스하기 위한 Oracle 네트워크 구성 (계속)

- 4) Enterprise Manager에서 **testorcl** Local Naming을 사용하여 **system** 유저와 **oracle_4U** 암호로 파트너의 orcl 데이터베이스에 액세스해 봅니다.

- a) Local Naming 페이지에서 **testorcl**을 선택한 다음 **Test Connection**을 누릅니다.

"Test Connection To Net Service Name: testorcl" 메시지가 나타납니다.

- b) Username에 **system**, Password에 **oracle_4U**를 입력한 다음 **Test**를 누릅니다.

Processing 페이지에 상태 정보가 표시됩니다. 이어서 성공 메시지가 나타납니다. 오류나 경고가 나타나면 이를 해결하십시오.

Test Connection To Net Service Name: testorcl

i Test Result
The test was successful. See log for details.

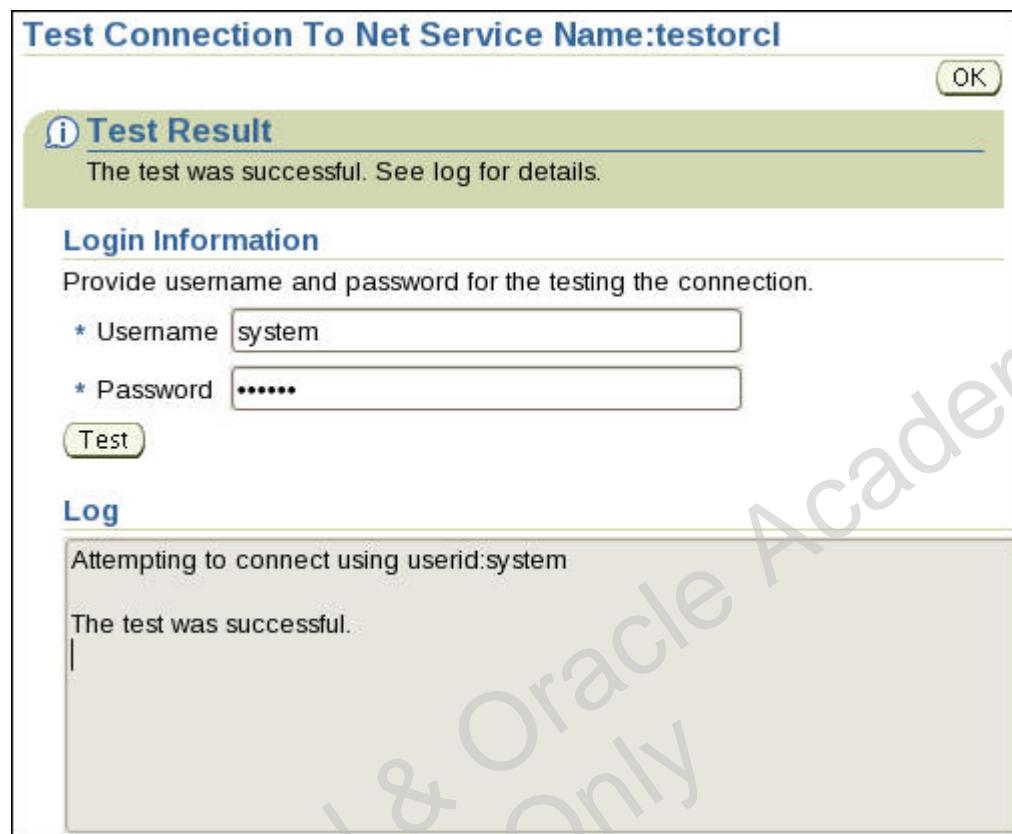
Login Information
Provide username and password for the testing the connection.

* Username

* Password

Test

Log
Attempting to connect using userid:system
The test was successful.



테스트가 완료된 다음 **OK**를 누릅니다.

연습 6-1: 다른 데이터베이스를 액세스하기 위한 Oracle 네트워크 구성 (계속)

- 5) SQL*Plus를 사용하여 네트워크 구성 변경 사항을 테스트합니다.
- system@testorcl**을 입력하고 암호를 묻는 메시지가 표시되면 **oracle_4U**를 입력합니다. 파트너의 정보를 보려면 v\$instance 테이블에서 instance_name 및 host_name 열을 선택합니다.
- a) oraenv를 실행하여 orcl 데이터베이스에 맞게 환경이 설정되었는지 확인합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

- b) 터미널 window에서 다음을 입력합니다.

```
$ sqlplus system@testorcl

SQL*Plus: Release 11.2.0.1.0 Production on
Fri Jul 10 11:07:11 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>
```

Oracle SQL*Plus window가 열립니다. 오류나 경고가 나타나면 이를 해결하십시오.

- c) SQL> 프롬프트에서 다음 명령을 입력합니다.

```
SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
-----
HOST_NAME
-----
orcl
edrsr25p1.us.oracle.com
```

연습 6-2: 대체 리스너 생성

이 연습에서는 Enterprise Manager를 사용하여 LISTENER2라는 두번째 리스너를 생성합니다.

- 1) LISTENER2라는 새 리스너를 생성합니다. 이 리스너에 대해 포트 1561을 사용합니다.
 - a) **SYSDBA** 풀의 **SYS** 유저로 Enterprise Manager에 로그인합니다. Database Instance – Home 페이지의 **General** 영역에서 **Listener** 링크를 누릅니다.
 - b) **Related Links** 영역에서 **Net Services Administration**을 누릅니다.
 - c) Net Services Administration 페이지의 **Administer** drop-down list에서 **Listeners**를 선택한 다음 **Go**를 누릅니다. Host Credentials의 username 및 암호에 각각 **oracle**을 입력한 다음 **Login**을 누릅니다.
 - d) 기존 리스너의 개요를 제공하는 **Listeners** 페이지에서 **Create** 버튼을 누릅니다. **Create Listener** 페이지가 나타납니다.
 - e) Listener Name에 **LISTENER2**를 입력한 다음 **Add**를 눌러 리스너 주소를 추가합니다.
 - f) 다음 값을 입력하거나 확인합니다.

옵션	값
Protocol	TCP/IP
Port	1561
Host	<컴퓨터의 호스트 이름. 예: edrsr12p1.us.oracle.com >

- g) **OK**를 누릅니다.
- h) **Static Database Registration** 탭을 누릅니다.



The screenshot shows the 'Create Listener' dialog box. At the top, there are tabs: General (selected), Authentication, Logging & Tracing, Static Database Registration (highlighted with a red box), and Other Services. Below the tabs, there is a field for 'Listener Name' containing 'LISTENER2'. Under the 'Addresses' section, it says 'Listener must have at least one address. If address is changed, listener will be stopped before applying changes.' There are 'Edit' and 'Remove' buttons, and an 'Add' button which is highlighted with a red box. A table below shows a single protocol entry: 'Protocol' is 'TCP/IP' and 'Protocol Details' show 'Host edrsr12p1.us.oracle.com' and 'Port 1561'.

연습 6-2: 대체 리스너 생성 (계속)

- i) Add 버튼을 눌러 새 리스너를 자신의 orcl 데이터베이스와 연결합니다.
- j) 다음 값을 입력합니다.

옵션	값
Service Name	orcl
Oracle Home Directory	/u01/app/oracle/product/11.2.0/dbhome_1
Oracle System Identifier (SID)	orcl



- k) OK를 눌러 데이터베이스 서비스를 추가합니다.
- l) OK를 눌러 LISTENER2리스너를 생성합니다.

Select	Name	Protocol Details	Status	Enterprise Manager Target
	LISTENER2	Protocol: TCP/IP Host: edrsr12p1.us.oracle.com Port: 1561	Stopped	Not a target

- 2) LISTENER2 리스너를 시작합니다.
 - a) LISTENER2 리스너와 Start/Stop Actions가 선택되었는지 확인한 다음 Go를 누릅니다.
 - b) Start/Stop 페이지에서 OK를 누릅니다.
View Details 링크와 함께 확인 메시지가 나타납니다.
 - c) 선택적으로 View Details 링크를 눌러 리스너 상태 정보를 검토하고 브라우저의 Back 아이콘을 사용하여 이전 페이지로 돌아갑니다.

연습 6-2: 대체 리스너 생성 (계속)

3) 새 리스너의 상태를 확인하고 새 리스너를 테스트합니다.

- a) **oraenv**를 실행하여 **orcl** 데이터베이스에 맞게 환경이 설정되었는지 확인합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

- b) 명령 프롬프트에서 다음 명령을 실행합니다.

```
$ lsnrctl status LISTENER2
```

- c) 간단한 연결(Easy Connect) 문자열을 사용하여 새 리스너로 데이터베이스에 연결합니다.

주: 이 연결 방법은 운영 환경에서는 권장되는 방법이 아닙니다. 이 방법은 간단한 강의실 환경에서 새로 생성된 리스너의 작동 여부를 입증하기 위해서 사용됩니다.

```
$ sqlplus hr/oracle_4U@your_ip_address:1561/orcl
```

새로 생성된 리스너를 통해 연결됩니다. 이 단계를 마친 후에는 SQL*Plus를 종료합니다.

- 4) 남은 과정에서는 이 새로운 LISTENER2가 필요하지 않으므로 이제 삭제할 수 있습니다.

```
$ lsnrctl stop LISTENER2
```

단원 7의 연습

현재 상황: 기존 저장 영역 구조 정보를 확인하고 INVENTORY 응용 프로그램에 대한 새 테이블스페이스를 생성해야 합니다. 또한 SYS 및 SYSTEM 계정을 사용하지 않고 모든 관리 작업을 수행할 데이터베이스 유저를 생성해야 합니다.

주: 유저 생성에 대해서는 설명되지 않았으므로, 이 연습을 위한 스크립트가 제공되었습니다.

Oracle Internal & Oracle Academy
Use Only

연습 7-1: 데이터베이스 저장 영역 구조 정보 보기

이 연습에서는 데이터베이스에 대한 기존 저장 영역 구조 정보를 검토합니다. 시작하기 전에 지금부터 DBA 작업에 사용하게 될 새 유저인 DBA1을 생성하는 스크립트를 실행합니다. Enterprise Manager에서 이 유저를 관리자 중 하나로 구성해야 합니다.

- 1) 이후에 사용할 디렉토리와 DBA1 유저를 생성하는 **lab_07_01_01.sh** 스크립트를 실행합니다. 이 스크립트는 **/home/oracle/labs**에 있습니다. DBA1의 암호는 oracle_4U입니다.
 - a) 이전 연습에서 열어 둔 터미널 window가 없으면 데스크톱을 마우스 오른쪽 버튼으로 누르고 **Open Terminal**을 선택하여 지금 하나를 엽니다.
 - b) orcl 데이터베이스에 연결할 수 있도록 환경 변수가 적절하게 설정되었는지 확인합니다.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

- c) 다음을 입력하여 현재 디렉토리를 labs 디렉토리로 변경합니다.

```
$ cd labs
```

- d) 다음 명령을 입력하여 DBA1 유저를 생성하는 스크립트를 실행합니다.

```
$ ./lab_07_01_01.sh
```

- e) 터미널 window를 그대로 열어 둡니다. 나중에 다시 사용하게 됩니다.

- 2) EM(Enterprise Manager) 오른쪽 상단 모서리에 있는 Setup 링크를 사용하여 **DBA1** 유저를 EM에서 관리 작업을 수행할 수 있는 유저로 정의합니다. 비SYS 유저가 구성된 경우 **SYS** 유저로 로그아웃하고 **DBA1** 유저로 로그인합니다. 별도의 지침이 없는 경우 DBA1 유저를 사용하여 이 작업의 나머지 부분을 수행합니다.

- a) EM window의 맨 오른쪽 상단 모서리에서 **Setup**을 누른 다음 Setup 페이지에서 **Administrators**를 선택합니다.



연습 7-1: 데이터베이스 저장 영역 구조 정보 보기 (계속)

- b) **Create**를 눌러 Administrators list에 DBA1 유저를 추가합니다. 이렇게 하면 DBA1 유저가 Enterprise Manager를 사용하여 관리 작업을 수행할 수 있습니다.

The screenshot shows the 'Administrators' page in Oracle Enterprise Manager. At the top, it says 'Administrators are database users who can login to Enterprise Manager to perform management tasks like set Blackouts, email notification schedules.' Below this, there is a search bar and a toolbar with buttons for View, Edit, Subscribe to Rules, Delete, and Create. The 'Create' button is highlighted with a red box. The main area shows a table with two columns: 'Select Name' and 'Access'. It lists three entries: SYS (Super Administrator), SYSMAN (Repository Owner), and SYSTEM (Super Administrator). The 'Access' column shows the privilege level for each user.

- c) Name에 **dba1**을 입력하고 Email Address는 비워 둡니다. Administrator Privilege로 Super Administrator를 선택한 다음 Review를 누릅니다.

The screenshot shows the 'Create Administrator: Properties' dialog. It has fields for Name (* Name: dba1), E-mail Address, and Administrator Privilege (Super Administrator). The 'Super Administrator' option is selected and highlighted with a red box. There is also a checked checkbox for 'Grant SELECT_CATALOG_ROLE'. At the bottom right, there are 'Cancel' and 'Review' buttons.

- d) Create Administrator dba1: Review 페이지에서 **Finish**를 누릅니다.
e) 비SYS 유저가 있으므로 오른쪽 상단 모서리에 있는 **Logout**을 누른 다음 **Login**을 누릅니다.
f) User Name에 **dba1**, Password에 **oracle_4U**를 입력하고 Connect As로 **SYSDBA**를 선택합니다. 그런 다음 **Login**을 누릅니다.
Database Home 페이지가 나타납니다.

- 3) Enterprise Manager를 사용하여 **EXAMPLE** 테이블스페이스에 대한 정보를 확인합니다. 다음 질문에 답하십시오.

- a) 질문 1: Warning 임계값에 도달하기 전까지 사용할 수 있는 여유 공간의 백분율은 얼마입니까?
i) Enterprise Manager에서 Server > Storage > **Tablespaces**를 선택합니다.
ii) **EXAMPLE** 테이블스페이스 이름을 누릅니다.
해답: 85%

The screenshot shows the 'Tablespace Full Metric Thresholds' dialog for the EXAMPLE tablespace. It states that the tablespace is using the database default space used thresholds. It displays two threshold values: 'Warning (%)' at 85 and 'Critical (%)' at 97. The 'Warning (%)' value is highlighted with a red box.

연습 7-1: 데이터베이스 저장 영역 구조 정보 보기 (계속)

- b) 질문 2: EXAMPLE 테이블스페이스에 몇 개의 세그먼트가 있습니까?
- Actions drop-down list에서 **Show Tablespace Contents**를 선택한 다음 **Go**를 누릅니다.
 - Show Tablespace Contents 페이지가 나타납니다.
- 해답: 420(답이 다를 수도 있음)*

Segments			
Search			
Segment Name	Type	Minimum Size (KB)	Minimum Extents
<input type="text"/> All Types <input type="button" value="Go"/>			
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.			
<input type="radio"/> Previous	1-10 of 420	<input type="radio"/> Next 10	
Segment Name	Type	Size (KB)	Extents
SH.CUSTOMERS	TABLE	12,288	27
PM.SYS_LOB0000073976C00034\$\$	LOBSEGMENT	5,120	20
SH.SUPPLEMENTARY_DEMOGRAPHICS	TABLE	4,096	19
PM.SYS_LOB0000073976C00054\$\$	LOBSEGMENT	4,096	19
OE.PRODUCT_DESCRIPTIONS	TABLE	3,072	18
SH.SALES.SALES_Q4_2001	TABLE PARTITION	2,048	17
SH.SALES.SALES_Q1_1999	TABLE PARTITION	1,024	16
SH.SALES.SALES_Q3_2001	TABLE PARTITION	1,024	16
SH.CUSTOMERS_PK	INDEX	1,024	16
SH.SALES.SALES_Q3_1999	TABLE PARTITION	960	15
<input type="radio"/> Previous	1-10 of 420	<input type="radio"/> Next 10	

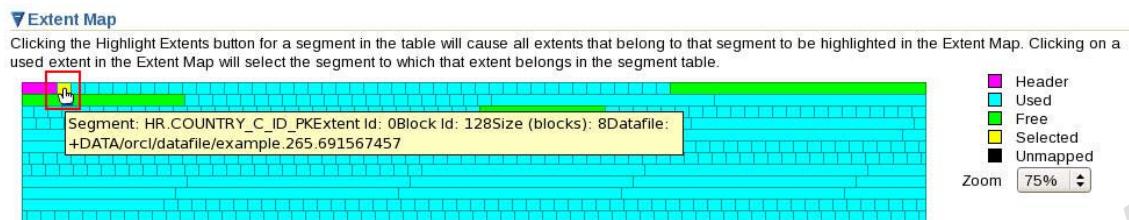
- c) 질문 3: EXAMPLE 테이블스페이스에서 어느 인덱스가 가장 많은 공간을 차지합니까?
- Search 영역의 Type drop-down list에서 INDEX를 선택한 다음 **Go**를 누릅니다.
 - Size 열은 정렬 열이며, 내림차순으로 정렬됩니다.

Segments			
Search			
Segment Name	Type	Minimum Size (KB)	Minimum Extents
<input type="text"/> INDEX <input type="button" value="Go"/>			
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.			
<input type="radio"/> Previous	1-10 of 78	<input type="radio"/> Next 10	
Segment Name	Type	Size (KB)	Extents
SH.CUSTOMERS_PK	INDEX	1,024	16
OE.PROD_NAME_IX	INDEX	512	8
OE.PRD_DESC_PK	INDEX	320	5
SH.CUSTOMERS_YOB_BIX	INDEX	192	3
SH.TIMES_PK	INDEX	128	2
<input type="radio"/> Previous	1-10 of 78	<input type="radio"/> Next 10	

해답: SH.CUSTOMERS_PK

연습 7-1: 데이터베이스 저장 영역 구조 정보 보기 (계속)

- d) 질문 4: 테이블스페이스에 물리적으로 처음 저장되는 세그먼트는 무엇입니까? 즉, 테이블스페이스 헤더 바로 뒤에 저장되는 세그먼트는 무엇입니까?
- 페이지 아래쪽으로 이동한 다음 Extent Map 레이블 왼쪽에 있는 더하기(+) 아이콘을 누릅니다.
 - 몇 초 후에 Extent 맵이 나타납니다. 맵 범례에서 분홍색은 테이블스페이스 헤더를 나타냅니다.
 - 다시 페이지 맨 위로 이동하여 Type drop-down list에서 All Types를 선택한 다음 Go를 누릅니다.
 - 테이블스페이스 헤더 Extent 바로 오른쪽에 있는 Extent를 누릅니다(선택된 경우 Extent가 노란색으로 바뀜). 세그먼트 위에서 커서를 이동하면 해당 위치에 저장된 세그먼트의 이름이 표시됩니다.



- v) 다시 페이지 맨 위로 이동하면 세그먼트가 다음을 가리키고 있습니다.

Highlight Extents	Segment Name	Type	Size (KB)	Extents
C	SH.COSTS.COSTS_Q1_2003	TABLE PARTITION	64	1
C	SH.COSTS.COSTS_Q2_2003	TABLE PARTITION	64	1
C	SH.COSTS.COSTS_Q3_2003	TABLE PARTITION	64	1
C	SH.COSTS.COSTS_Q4_2003	TABLE PARTITION	64	1
C	HR.REG_ID_PK	INDEX	64	1
➡	HR.COUNTRY_C_ID_PK	INDEX	64	1
C	HR.LOC_ID_PK	INDEX	64	1
C	HR.LOC_CITY_IX	INDEX	64	1
C	HR.LOC_STATE_PROVINCE_IX	INDEX	64	1
C	HR.LOC_COUNTRY_IX	INDEX	64	1

답: HR.COUNTRY_C_ID_PK

연습 7-2: 테이블스페이스 생성

이 연습에서는 이후 연습에서 사용되는 Inventory 테이블스페이스를 생성합니다.

- 1) 크기가 **5MB**인 **INVENTORY**라는 새 LMT(로컬관리방식 테이블스페이스)를 생성합니다.
 - a) Enterprise Manager에서 **Server > Storage > Tablespaces**를 선택합니다.
 - b) **Create**를 누릅니다.
 - c) 테이블스페이스 이름으로 **INVENTORY**를 입력하고 Extent Management가 **Locally Managed**, Type이 **Permanent**, Status가 **Read Write**이며 Use Bigfile tablespace가 선택되지 않았는지 확인합니다.
 - d) Datafiles 영역에서 **Add**를 누릅니다.
 - e) Add Datafile 페이지에서 Storage Type으로 **Automatic Storage Management**을 선택하고 DiskGroup으로 **DATA**가 선택되었는지 확인한 다음 File Size로 **5MB**를 입력합니다. 그런 다음 **Continue**를 누릅니다.

The screenshot shows the 'Add Datafile' dialog box. Under the 'Storage Type' section, 'Automatic Storage Management' is selected. The 'DiskGroup' dropdown shows 'DATA'. The 'Template' dropdown shows '<Default>'. The 'Tablespace' dropdown shows 'INVENTORY'. The 'File Size' input field contains '5' followed by a 'MB' unit selector. Below it, the 'Reuse Existing File' checkbox is unchecked. In the 'Storage' section, the 'Automatically extend datafile when full (AUTOEXTEND)' checkbox is unchecked. The 'Increment' input field is empty. The 'Maximum File Size' section has two options: 'Unlimited' (selected) and 'Value' (radio button unselected), with 'Value' input field showing '0' and 'MB' unit selector.

- f) **Storage** 탭을 누르고 Extent Allocation 및 Segment Space Management가 **Automatic**이고 Compression Options가 **Disabled**이고 Logging이 **Yes**로 설정되었는지 확인합니다.

연습 7-2: 테이블스페이스 생성 (계속)

- g) General 탭을 누르고 설정을 검토합니다.

Create Tablespace

Show SQL Cancel OK

i Information
Modification to the datafile will not take effect until you click "OK" button.

General Storage

* Name INVENTORY

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

Datafiles
 Use bigfile tablespace
Tablespace can have only one datafile with no practical size limit.
Add

Select Name	Directory	Size (MB)
<input checked="" type="radio"/> <ASM Generated>	+DATA/	5.00

- h) Show SQL을 눌러 실행할 SQL을 확인한 다음 Return을 누릅니다.

- i) OK를 누르면 생성 성공 메시지가 나타납니다.

연습 7-2: 테이블스페이스 생성 (계속)

- 2) DBA1 유저로 **lab_07_02_02.sql** 스크립트를 실행하여 INVENTORY 테이블스페이스에서 X라는 테이블을 생성하고 채웁니다. 결국 어떤 오류가 나타납니까?
- a) 터미널 window에서 labs 디렉토리로 이동합니다. 아직 수행하지 않은 경우에는 oraenv를 사용하여 orcl 데이터베이스에 맞게 환경을 설정해야 합니다.

```
$ cd labs
```

- b) dba1 유저와 **oracle_4U** 암호로 SQL*Plus에 로그인하고 **lab_07_02_02.sql** 스크립트를 실행합니다.
주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus dba1
```

```
SQL*Plus: Release 11.2.0.1.0 Production on
Wed Jul 8 12:06:50 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> @lab_07_02_02.sql
```

- c) 테이블을 확장할 수 없음을 의미하는 ORA-01653 오류가 나타납니다. 삽입되는 모든 행을 수용하기에 충분한 공간이 없습니다.

```
...
SQL> insert into x select * from x
  2  /
1024 rows created.

SQL> insert into x select * from x
  2  /
insert into x select * from x
*
ERROR at line 1:
ORA-01653: unable to extend table DBA1.X by 128 in tablespace
INVENTORY
```

연습 7-2: 테이블스페이스 생성 (계속)

```
SQL> commit  
2 /  
  
Commit complete.  
  
SQL> quit  
Disconnected from Oracle Database 11g Enterprise Edition  
Release 11.1.0.6.0 - Production  
With the Partitioning, OLAP, Data Mining and Real  
Application Testing options
```

- 3) Enterprise Manager window로 이동하여 **INVENTORY** 테이블스페이스에 대해 사용할 수 있는 공간의 양을 늘립니다. 교육을 위해 여기에서는 두 가지 다른 방법을 사용하여 이 작업을 수행하게 됩니다. 먼저 현재 데이터 파일의 크기를 **40MB**로 늘립니다. 그런 다음 동일 테이블스페이스에 대해 ASM 및 비ASM 데이터 파일이 모두 존재할 수 있는지 확인하기 위해 파일 시스템 저장 영역을 사용하여 두 번째 데이터 파일을 추가합니다. 이 두 번째 데이터 파일은 크기가 **30MB**여야 합니다. 두 기술 모두 지원하는 SQL 문을 보려면 SQL 표시 기능을 사용합니다.
- Server> Storage > **Tablespaces**를 선택합니다.
 - INVENTORY 테이블스페이스를 선택한 다음 **Edit**를 누릅니다.
 - Datafiles 영역에서 **Edit**를 누릅니다.
 - File Size를 5MB에서 **40MB**로 변경합니다.
 - Continue**를 눌러 General 탭 페이지로 돌아갑니다.
 - Show SQL**을 눌러 실행할 SQL을 확인합니다. 이 명령문은 ALTER DATABASE 문입니다. **Return**을 누릅니다.

Show SQL **Return**
ALTER DATABASE DATAFILE '+DATA/orcl/datafile/inventory.268.693647553' RESIZE 40M

- Datafiles 영역에서 **Add**를 누릅니다.
- Storage Type으로 **File System**을 선택합니다. File Name으로 **inventory02.dbf**를 입력합니다. File Directory에 **/u01/app/oracle/oradata/orcl**을 입력합니다. File Size에 **30MB**를 입력합니다.
주: 이 디렉토리는 앞에서 실행한 스크립트로 생성되었습니다.
- Continue**를 눌러 General 탭 페이지로 돌아갑니다.
- Show SQL**을 눌러 실행할 SQL을 확인합니다. 이 명령문은 ALTER DATABASE 문입니다. **Return**을 누릅니다.

Show SQL **Return**
ALTER TABLESPACE "INVENTORY" ADD DATAFILE '/u01/app/oracle/oradata
/orcl/inventory02.dbf' SIZE 30M

연습 7-2: 테이블스페이스 생성 (계속)

- k) **Apply**를 누릅니다.
- l) 이제 INVENTORY 테이블스페이스에 대해 두 개의 데이터파일이 있습니다.
하나는 ASM 저장 영역을 사용하고 다른 하나는 파일 시스템(비ASM) 저장 영역을 사용합니다.

Datafiles				
		Directory	Size (MB)	Used (MB)
<input checked="" type="radio"/>	inventory02.dbf	/u01/app/oracle/oradata/orcl/	30.00	-34.00
<input type="radio"/>	inventory_268.693647553	+DATA/orcl/datafile/	40.00	-24.00

- 4) 터미널 window로 돌아가서 **lab_07_02_04.sql** 스크립트를 실행합니다. 이 스크립트는 테이블을 삭제하고 이전에 공간 오류를 반환한 원래 스크립트를 재실행합니다.
 - a) 터미널 window로 이동합니다.
 - b) **dba1** 유저와 **oracle_4U** 암호로 SQL*Plus에 로그인하고 **lab_07_02_04.sql** 스크립트를 실행합니다.
주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus dba1

SQL*Plus: Release 11.2.0.1.0 Production on
Wed Jul 8 12:06:50 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> @lab_07_02_04.sql
```

- c) 동일한 수의 행 삽입을 시도해도 증가된 테이블스페이스 크기로 인해 오류가 발생하지 않습니다.

연습 7-2: 테이블스페이스 생성 (계속)

- 5) 터미널 window에서 **dba1** 유저로 SQL*Plus를 사용해서 **lab_07_02_05.sql** 스크립트를 실행하여 뒷부분의 연습 세션을 위해 테이블스페이스를 정리합니다.
주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus dba1

SQL*Plus: Release 11.2.0.1.0 Production on
Wed Jul 8 12:06:50 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> @lab_07_02_05.sql
```

Oracle Internal & Oracle Academy
Use Only

단원 8의 연습

현재 상황: 새 인적 자원 관리(HR) 부서 관리자인 Jenny Goodman에게 유저 계정을 생성해 주어야 합니다. HR 부서에는 David Hamby와 Rachel Pandya라는 두 명의 신입 직원도 있습니다. 세 명 모두 orc1 데이터베이스에 로그인하고 HR.EMPLOYEES 테이블에서 데이터를 선택하고 테이블의 레코드를 갱신할 수 있어야 합니다. 관리자는 또한 새 사원 레코드를 삽입하고 삭제할 수 있어야 합니다. 새 유저가 퇴근 시에 깜박 잊고 로그아웃하지 않은 경우 15분이 지나면 자동으로 로그아웃되어야 합니다. 이 연습에서는 또한 설치하고 있는 재고 관리 응용 프로그램에 대해 새 유저 계정을 생성해야 합니다.

Oracle Internal & Oracle Academy
Use Only

연습 8-1: 프로파일 생성 및 사용

이 연습에서는 새 재고 관리 응용 프로그램을 소유할 INVENTORY 유저를 생성합니다. 유저의 Idle 시간을 제한하는 프로파일을 생성합니다. 유저가 Idle 상태이거나 깜박 잊고 로그아웃하지 않은 경우 15분이 지나면 유저 세션이 종료됩니다.

- 필수 작업: **lab_08_01_01.sh** 스크립트(/home/oracle/labs 디렉토리에 있음)를 검토하고 실행하여 다음 연습에서 사용할 **INVENTORY** 유저(암호 **oracle_4U**)를 생성합니다.

- 터미널 window에서 다음을 입력합니다.

```
$ cd $HOME/labs
$ cat lab_08_01_01.sh
# Oracle Database 11g: Administration Workshop I
# Oracle Server Technologies - Curriculum Development
#
# ***Training purposes only***
#***Not appropriate for production use***
#
# Start this script as OS user: oracle
#   This script creates the INVENTORY schema user
#   The DROP command fails the first time
#   you execute the script.
#   The error can be ignored.

cd ~/labs

. set_db.sh

sqlplus / as sysdba << EOF
drop user inventory cascade;

create user inventory identified by oracle_4U
default tablespace inventory;

grant connect, resource to inventory;

exit;
EOF
$ ./lab_08_01_01.sh

SQL*Plus: Release 11.2.0.1.0 Production on
Wed Jul 8 13:00:36 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
```

연습 8-1: 프로파일 생성 및 사용 (계속)

```
and Real Application Testing options

SQL> SQL> drop user inventory cascade
      *
ERROR at line 1:
ORA-01918: user 'INVENTORY' does not exist

SQL> SQL> 2
User created.

SQL> SQL>
Grant succeeded.

SQL> 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
$
```

- 2) Idle 시간을 15분만 허용하는 **HRPROFILE**이라는 프로파일을 생성합니다.
 - a) **orcl** 데이터베이스에 대해 **SYSDBA** 률의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
 - b) **Server** 탭을 누른 다음 Security 섹션에서 **Profiles**를 누릅니다.
 - c) **Create** 버튼을 누릅니다.
 - d) Name 필드에 **HRPROFILE**을 입력합니다.
 - e) Idle Time (Minutes) 필드에 **15**를 입력합니다.
 - f) 다른 필드는 모두 **DEFAULT**로 설정된 상태로 둡니다.
 - g) **Password** 탭을 누르고 현재 모두 DEFAULT로 설정된 Password 옵션을 검토합니다.
 - h) 선택적으로 **Show SQL** 버튼을 눌러 기본 SQL 문을 검토한 다음 **Return**을 누릅니다.
 - i) 마지막으로 **OK**를 눌러 프로파일을 생성합니다.
- 3) 프로파일 제한이 적용되도록 **RESOURCE_LIMIT** 초기화 파라미터를 **TRUE**로 설정합니다.
 - a) **Server** 탭을 누른 다음 Database Configuration 섹션에서 **Initialization Parameters**를 누릅니다.
 - b) Name 필드에 **resource_limit**을 입력한 다음 **Go**를 누릅니다.
 - c) Value drop-down list에서 **TRUE**를 선택한 다음 **Apply**를 누릅니다.

연습 8-2: 룰 생성

이 연습에서는 다음 연습에서 사용할 **HRCLERK** 및 **HRMANAGER** 룰을 생성합니다.

- 1) **HR.EMPLOYEES** 테이블에 대해 **SELECT** 및 **UPDATE** 권한이 있는 **HRCLERK**라는 룰을 생성합니다.
 - a) Server 탭을 누른 다음 Security 섹션에서 Roles를 누릅니다.
 - b) Create 버튼을 누릅니다.
 - c) Name 필드에 **HRCLERK**을 입력합니다. 이 룰은 인증되지 않았습니다.
 - d) Object Privileges 탭을 누릅니다.
 - e) Select Object Type drop-down list에서 Table을 선택한 다음 Add를 누릅니다.
 - f) Select Table Objects 필드에 **HR.EMPLOYEES**를 입력합니다.
 - g) **SELECT** 및 **UPDATE** 권한을 Selected Privileges 상자로 이동합니다. OK를 누릅니다.
 - h) Show SQL 버튼을 눌러 기본 SQL 문을 검토합니다.

Show SQL

```
CREATE ROLE "HRCLERK" NOT IDENTIFIED
GRANT SELECT ON "HR"."EMPLOYEES" TO "HRCLERK"
GRANT UPDATE ON "HR"."EMPLOYEES" TO "HRCLERK"
```

- i) Return을 누른 다음 OK를 눌러 룰을 생성합니다.
- 2) **HR.EMPLOYEES** 테이블에 대해 **INSERT** 및 **DELETE** 권한이 있는 **HRMANAGER**라는 룰을 생성합니다. **HRCLERK** 룰을 **HRMANAGER** 룰에 부여합니다.
 - a) Server 탭을 누른 다음 Security 섹션에서 Roles를 누릅니다.
 - b) Create를 누릅니다.
 - c) Name 필드에 **HRMANAGER**을 입력합니다. 이 룰은 인증되지 않았습니다.
 - d) Object Privileges 탭을 누릅니다.
 - e) Select Object Type drop-down list에서 Table을 선택한 다음 Add를 누릅니다.
 - f) Select Table Objects 필드에 **HR.EMPLOYEES**를 입력합니다.
 - g) **INSERT** 및 **DELETE** 권한을 Selected Privileges 상자로 이동합니다. OK를 누릅니다.

연습 8-2: 룰 생성 (계속)

- h) Roles 탭을 누른 다음 Edit List를 누릅니다.
- i) **HRCLERK** 를을 Selected Roles 상자로 이동한 다음 OK를 누릅니다.
- j) Show SQL 버튼을 눌러 기본 SQL 문을 검토합니다.

Show SQL Return

```
CREATE ROLE "HRMANAGER" NOT IDENTIFIED
GRANT DELETE ON "HR"."EMPLOYEES" TO "HRMANAGER"
GRANT INSERT ON "HR"."EMPLOYEES" TO "HRMANAGER"
GRANT "HRCLERK" TO "HRMANAGER"
```

- k) Return을 누른 다음 OK를 눌러 룰을 생성합니다.

Oracle Internal & Oracle Academy
Use Only

연습 8-3: 유저 생성 및 구성

이 연습에서는 다음과 같은 유저들을 생성하고 적합한 프로파일 및 룰을 유저에 할당합니다.

이름	Username	설명
David Hamby	DHAMBY	새 HR Clerk
Rachel Pandya	RPANDYA	새 HR Clerk
Jenny Goodman	JGOODMAN	새 HR Manager

- 1) 새 HR 직원인 David Hamby에 대한 계정을 생성합니다.
 - a) Server 탭을 누른 다음 Security 섹션에서 **Users**를 누릅니다.
 - b) **Create**를 누르고 Name 필드에 **DHAMBY**를 입력합니다.
 - c) Profile에 **HRPROFILE**을 선택합니다.
 - d) Password Authentication을 선택하고 암호로 **newuser**를 입력합니다. **Confirm Password** 필드에도 이 값을 입력합니다. David가 처음 로그인할 때 암호를 변경하도록 **Expire Password now** 체크 박스를 선택합니다.
 - e) **Roles** 탭을 누릅니다. CONNECT 풀이 자동으로 유저에게 할당되었습니다.
 - f) Edit List를 누르고 **HRCLERK** 풀을 Selected Roles 상자로 이동하여 **HRCLERK** 풀을 추가합니다. **OK**를 눌러 Modify Roles window를 닫습니다.
 - g) **OK**를 다시 눌러 유저를 생성합니다.
- 2) 또 다른 새 HR 직원인 Rachel Pandya에 대한 계정을 생성합니다. 위의 1단계에서 표시된 단계를 반복하지만 이번에는 username으로 **RPANDYA**를 사용합니다.
- 3) 새 HR 관리자인 Jenny Goodman에 대한 계정을 생성합니다. 1단계에서 표시된 단계를 반복하지만 username으로 **JGOODMAN**을 사용하고 **HRCLERK** 풀 대신 **HRMANAGER** 풀을 선택합니다.
 - a) **Show SQL** 버튼을 눌러 기본 SQL 문을 검토합니다.

Show SQL **Return**

```
CREATE USER "JGOODMAN" PROFILE "HRPROFILE" IDENTIFIED BY *****
PASSWORD EXPIRE ACCOUNT UNLOCK
GRANT "CONNECT" TO "JGOODMAN"
GRANT "HRMANAGER" TO "JGOODMAN"
```

- b) **Return**을 누른 다음 **OK**를 눌러 유저를 생성합니다.

연습 8-3: 유저 생성 및 구성 (계속)

- 4) SQL*Plus에서 새 유저를 테스트합니다. orcl 데이터베이스에 DHAMBY 유저로 연결합니다. 새 암호로 **oracle_4U**를 사용합니다. **HR.EMPLOYEES** 테이블에서 **EMPLOYEE_ID=197**라는 값이 있는 행을 선택합니다. 그런 다음 이 행의 삭제를 시도합니다. ("insufficient privileges" 오류 메시지가 나타나야 합니다.)

a) 터미널 window에서 다음을 입력합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus dhamby
```

또는 이미 SQL*Plus가 실행 중인 경우 CONNECT 명령을 사용합니다.

SQL*Plus에서 dhamby로 다시 연결하는 경우 로그인 및 암호 변경 세션의 상태는 다음과 유사합니다.

```
SQL> CONNECT dhamby
Enter password: newuser <<<Password does not appear on screen
ERROR:
ORA-28001: the password has expired

Changing password for dhamby
New password: oracle_4U <<<Password does not appear on screen
Retype new password: oracle_4U <<<Password does not appear on
screen
Password changed

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>
```

b) **HR.EMPLOYEES** 테이블에서 **EMPLOYEE_ID=197**의 급여를 선택합니다.

```
SQL> SELECT salary FROM hr.employees WHERE EMPLOYEE_ID=197;
      SALARY
      -----
      3000
```

c) 이제 **hr.employees** 테이블에서 동일한 레코드를 삭제해 봅니다.

```
SQL> DELETE FROM hr.employees WHERE EMPLOYEE_ID=197;
DELETE FROM hr.employees WHERE EMPLOYEE_ID=197
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

연습 8-3: 유저 생성 및 구성 (계속)

- 5) **JGOODMAN** 유저로 테스트를 반복합니다. 새 암호로 **oracle_4U**를 사용합니다. 행을 삭제한 후 롤백을 실행하면 원래의 107 행이 그대로 유지되어야 합니다.

- a) **orcl** 데이터베이스에 **JGOODMAN** 유저로 연결합니다.

```
SQL> connect jgoodman
Enter password:
ERROR:
ORA-28001: the password has expired
<Change the password to oracle_4U as shown above>
```

- b) HR.EMPLOYEES 테이블에서 EMPLOYEE_ID=197라는 값이 있는 행을 선택합니다.

```
SQL> SELECT salary FROM hr.employees WHERE EMPLOYEE_ID=197;

      SALARY
-----
      3000
```

- c) 이제 HR.EMPLOYEES 테이블에서 동일한 행을 삭제합니다.

```
SQL> DELETE FROM hr.employees WHERE EMPLOYEE_ID=197;

1 row deleted.
```

- d) 이 과정은 단지 테스트이기 때문에 삭제 작업을 롤백합니다.

```
SQL> rollback;

Rollback complete.
```

- e) 이 테이블에 여전히 107개의 행이 있는지 확인합니다.

```
SQL> SELECT COUNT(*) FROM hr.employees;

      COUNT(*)
-----
      107

SQL>
```

연습 8-3: 유저 생성 및 구성 (계속)

질문 1: 행은 삭제 후 어디에 저장되었습니까?

해답: 언두 테이블스페이스에 저장되었습니다.

질문 2: 새 유저를 생성할 때 기본 테이블스페이스나 임시 테이블스페이스를 선택하지 않았습니다. 새 유저가 사용할 테이블스페이스를 어떻게 결정합니까?

해답: 시스템에서 정의한 기본 영구 테이블스페이스 및 임시 테이블스페이스를 사용합니다.

질문 3: 새 유저에게 CREATE SESSION 시스템 권한을 부여하지 않았지만 새 유저가 모두 데이터베이스에 연결할 수 있습니다. 왜 그럴까요?

해답: Enterprise Manager가 자동으로 CONNECT 룰을 새 유저에게 할당하므로 CREATE SESSION이 해당 룰에 포함됩니다.

- 6) SQL*Plus를 사용하여 **RPANDYA** 유저로 **orcl** 데이터베이스에 연결합니다. 암호를 **oracle_4U**로 변경합니다. (RPANDYA로 처음 연결하는 것으로 암호를 변경해야 합니다.) 다음 단원이 진행되는 동안이나 업무가 끝날 시점에 RPANDYA를 연결된 상태로 둡니다. HPROFILE은 세션이 15분 이상 비활성화된 유저를 자동으로 로그아웃하도록 지정합니다. **HR.EMPLOYEES** 테이블에서 다시 선택을 시도하여 해당 유저가 자동으로 로그아웃되었는지 확인해 보십시오.

```
ERROR at line 1:  
ORA-02396: exceeded maximum idle time, please connect again
```

단원 9의 연습

현재 상황: Help Desk에서 방금 HR 담당자인 Susan Mavris로부터 데이터베이스가 "고정"되어 있다고 불평하는 전화를 받았습니다. 원인을 조사해 보니 Susan Mavris가 John Chen의 사원 레코드를 새 전화 번호로 갱신하려고 했다는 사실을 알았습니다. 그러나 그녀가 새 데이터를 입력했을 때 세션이 고정되어 더 이상 아무것도 수행할 수 없었습니다. SQL 스크립트 파일은 /home/oracle/labs 디렉토리에 제공되어 있습니다.

Oracle Internal & Oracle Academy
Use Only

연습 9-1: Lock 충돌 해결

이 연습에서는 두 개의 개별 SQL*Plus 세션을 사용하여 lock 충돌을 일으킵니다. Enterprise Manager를 사용하여 lock 충돌의 원인을 감지한 후 충돌을 해결합니다. 유저 편의를 위해 lock 충돌을 일으키는 SQL 코드는 이 연습에서 실행하는 스크립트에 제공되었습니다.

- 1) **lab_09_01_01.sql** 스크립트를 실행하여 문제의 행에 대해 커밋되지 않은 갱신을 수행합니다. 이 스크립트는 먼저 이 연습에 포함된 유저(smaovris 및 ngreenberg)를 생성하고 이러한 새 유저에게 hr.employee 테이블에 대한 액세스 권한을 부여하는 hremployee 룰을 생성합니다. 그런 다음 ngreenberg 유저로 SQL*Plus에 로그인하고 hr.employee 테이블에서 갱신을 수행합니다. 이 스크립트는 커밋을 수행하지 않으므로 이 세션에서 이 갱신 작업을 커밋되지 않은 상태로 둡니다.
 - a) oraenv를 실행하여 orcl 데이터베이스에 맞게 환경이 구성되었는지 확인합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

- b) 다음을 입력하여 스크립트를 실행합니다. 스크립트 실행이 완료되면 커밋되지 않은 갱신이 수행되었다는 메시지가 나타납니다.

```
$ sqlplus dba1

SQL*Plus: Release 11.2.0.1.0 Production on
Thu Jul 9 03:57:42 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> @lab_09_01_01.sql
Creating users...
... [redacted] ... 일부 출력은 공간을 줄이기 위해 표시되지 않음 ...
Connecting as ngreenberg ...
Connected.
```

연습 9-1: Lock 충돌 해결 (계속)

```
SQL> show user
USER is "NGREENBERG"
SQL> update hr.employees set phone_number='650.555.1212'
where employee_id = 110 ;

1 row updated.

SQL> prompt User "ngreenberg" made an update and left it
uncommitted in this session.
User "ngreenberg" made an update and left it uncommitted
in this session.
SQL>
SQL>
SQL>
```

- c) 현재 있는 상태로 세션 연결을 유지합니다. 아직 종료하지 마십시오.
- 2) 별도의 터미널 window에서 **lab_09_01_02.sql** 스크립트를 실행하여 별도의 세션에서 동일한 행을 갱신해 봅니다. 계속하기 전에 "Update is being attempted now" 메시지를 확인합니다. 세션이 "정지"된 것처럼 보이더라도 신경 쓰지 마십시오. 이는 단지 문제 해결을 위해 설정한 상황입니다.
- a) 또 다른 명령 셸을 시작할 터미널 window를 열고 다음을 입력하여 두 번째 스크립트를 실행합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus dba1

SQL*Plus: Release 11.2.0.1.0 Production on
Thu Jul 9 04:04:47 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
```

연습 9-1: Lock 충돌 해결 (계속)

```
SQL> @lab_09_01_02.sql
Sleeping for 20 seconds to ensure first process gets the
lock first.

PL/SQL procedure successfully completed.

Sleep is finished.
Connected.
USER is "SMAVRIS"
Update is being attempted now.
```

- b) 이 세션은 정지된 것으로 보입니다. 이 세션을 있는 그대로 두고 다음 단계로 이동합니다.
- 3) Enterprise Manager에서 Performance 페이지의 **Blocking Sessions** 링크를 눌러 어느 세션이 Lock 충돌을 유발하고 있는지 알아냅니다.
- Enterprise Manager에서 **Performance** 페이지를 누릅니다.
 - Additional Monitoring Links** 영역에서 **Blocking Sessions**를 누릅니다. 출력은 다음과 같아야 합니다.

Blocking Sessions											
Page Refreshed Jul 9, 2009 4:13:46 AM GMT+07:00 Refresh											
View Session Kill Session Expand All Collapse All											
Select Username	Sessions Blocked	Session ID	Serial Number	Wait SQL ID	Wait Class	Wait Event	P1 Value	P2 Value	P3 Value	Seconds In Wait	
Blocking Sessions											
NGREENBERG	1	51	7460		Idle	SQL*Net message from client	16508152321	0		862	
SMAVRIS	0	52	4319 6smgtv6h8958b	Application	eng: TX - row lock contention	6smgtv6h8958b	1415053318	327697	778	383	

- 4) 차단 중인 세션에서 실행한 마지막 SQL 문은 무엇입니까?
- NGREENBERG** 세션을 선택한 다음 **View Session**을 누릅니다.
 - Previous SQL**이라는 해시 값 링크를 누릅니다.

Session Details: 51 (NGREENBERG)											
Collected From Target Jul 9, 2009 4:21:57 AM GMT+07:00											
View Data Real Time: 15 Second Refresh Refresh Kill Session Enable SQL Trace											
General				Activity				Statistics			
Server				Client				Application			
Current Status: INACTIVE Serial Number: 7460 DB User Name: NGREENBERG OS Process ID: 7129 Login Time: Jul 9, 2009 3:59:24 AM Login Duration: 22:34 (mm:ss) Connection Type: DEDICATED Type: USER Resource Consumer Group: Unavailable				OS User Name: oracle OS Process ID: 6962 Host: edrsr12p1.us.oracle.com Terminal: pts/2 Current Client ID: Unavailable Current Client Info: Unavailable				Current SQL: None Current SQL Command: UNKNOWN Previous SQL: 6smgtv6h8958b Last Call Duration: 22:34 (mm:ss) SQL Trace: DISABLED Current SQL Trace Level: 1 Trace With Wait Information: DISABLED Trace With Bind Information: DISABLED Open Cursors: 35 Program: sqlplus@edrsr12p1.us.oracle.com (TNS V1-V3) Service: SYS\$USERS Current Module: SQL*Plus Current Action: Unavailable			

연습 9-1: Lock 충돌 해결 (계속)

- c) 가장 최근에 실행된 SQL을 확인합니다.

SQL Details: 6smgtv6h8958b

Switch to SQL ID Go View Data Real Time: Manual Refresh Refresh SQL Worksheet Schedule SQL Tuning Advisor SQL Repair Advisor

Text

```
update hr.employees set phone_number='650.555.1212' where employee_id = 110
```

- 5) 차단 중인 세션을 종료함으로써 신고한 유저의 불편이 해소되도록 충돌을 해결합니다. 충돌을 해결하는 SQL 문은 무엇입니까?
- 브라우저의 **Back** 버튼을 누릅니다.
 - 이제 Session Details: NGREENBERG 페이지에서 **Kill Session**을 누릅니다.
 - Options를 **Kill Immediate**로 설정된 상태로 두고 **Show SQL**을 눌러 세션을 종료하기 위해 실행할 명령문을 확인합니다.
- 주:** 세션과 일련 번호는 대부분 여기에 나오는 내용과 다릅니다.

Show DDL

Return

```
ALTER SYSTEM KILL SESSION '51,7460' IMMEDIATE
```

- Return을 누른 다음 Yes를 눌러 KILL SESSION 명령을 실행합니다.
- SQL*Plus 명령 window로 돌아가서 SMAVRIS의 갱신이 성공적으로 완료되었는지 확인합니다. 성공 메시지가 나타나려면 몇 초 걸릴 수 있습니다.

```
USER is "SMAVRIS"  
Update is being attempted now.  
  
1 row updated.  
  
Update is completed.  
SQL>
```

- 7) NGREENBERG 세션에서 SQL Select 문을 실행해 봅니다. 어떤 결과가 발생합니까?

```
SQL> SELECT sysdate from dual;  
SELECT sysdate from dual  
*  
ERROR at line 1:  
ORA-03135: connection lost contact  
Process ID: 7129  
Session ID: 51 Serial number: 7460  
  
SQL>
```

해답: 세션의 연결이 해제되었습니다.

exit를 입력하여 모든 열린 SQL 세션을 닫은 다음 터미널 window를 닫습니다.

단원 10의 연습

현재 상황: 조직의 업무 유저와 관리자측이 Flashback 요구를 지원하기 위해 오라클 데이터베이스에서 48시간의 언두 Retention을 두기로 결정했습니다. 여러분의 임무는 이러한 요구 사항을 지원하도록 orcl 데이터베이스를 구성하는 것입니다.

Oracle Internal & Oracle Academy
Use Only

연습 10-1: 언두 데이터 관리

이 연습에서는 먼저 언두와 관련한 시스템 작업을 확인한 다음 Flashback 작업에 48시간 Retention을 지원하도록 orcl 데이터베이스를 구성합니다.

- 1) **DBA1** 유저로 로그인한 Enterprise Manager에서 언두 관련 시스템 작업을 확인합니다.
 - a) **Server** 탭 페이지를 누르고 Database Configuration 섹션에서 **Automatic Undo Management**를 선택합니다.
 - b) **System Activity** 탭 페이지를 누릅니다.

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](#)

System Activity During Analysis Period

Selected Analysis Time Period	Jul 2, 2009 5:00:00 AM GMT+07:00 To Jul 9, 2009 5:00:00 AM GMT+07:00
Longest Running Query (minutes)	18.0
Average Undo Generation Rate (KB/minute)	75.0
Maximum Undo Generation Rate (KB/minute)	1,385.0
Queries failed due to low Retention	0
Transactions failed due to small Undo Tablespace	0

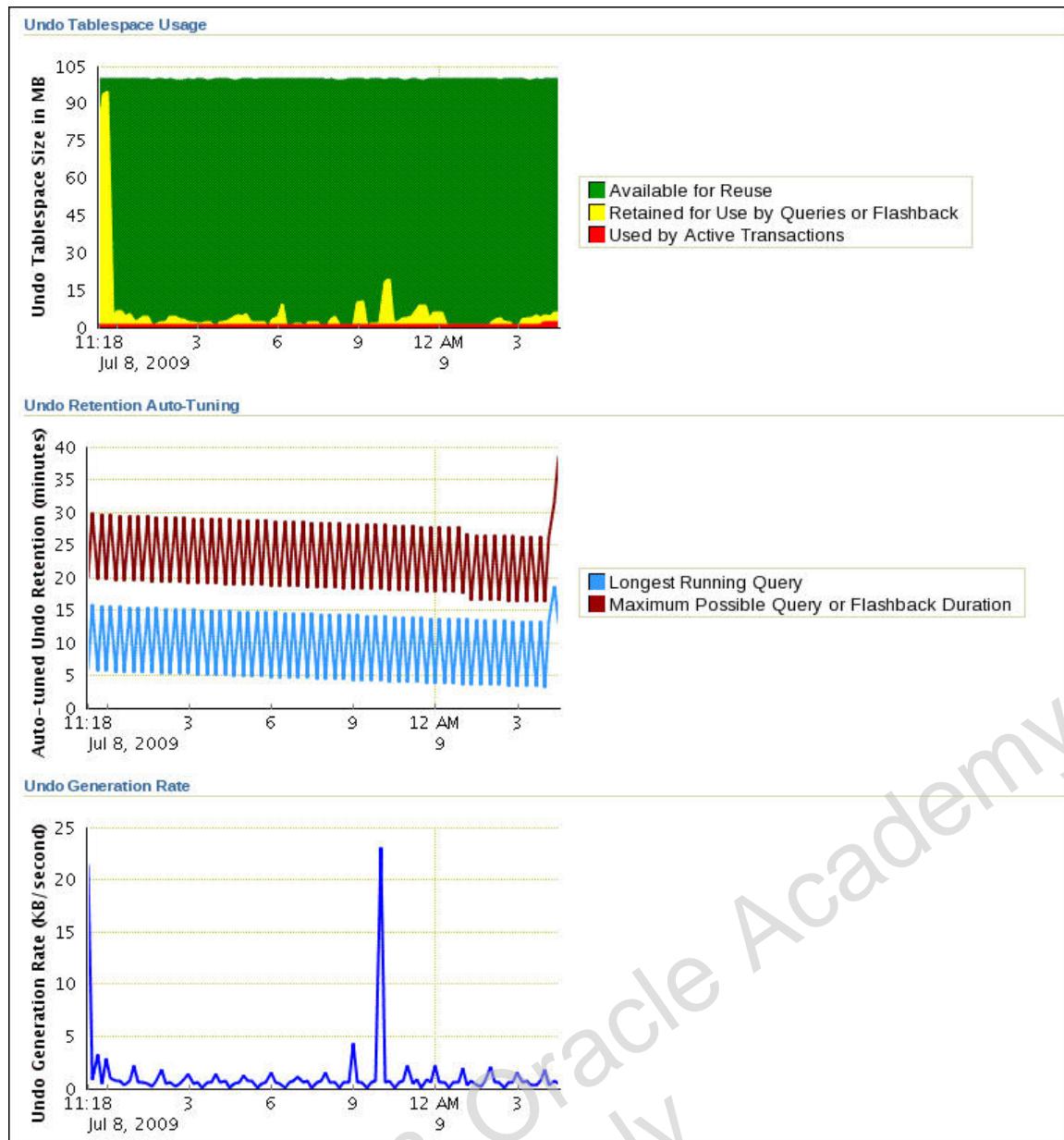
[Show Graph](#)

주: 분석 기간과 이 기간 동안의 시스템 작업에 따라 모든 분석 스크린샷에서 정보는 다르게 나타납니다.

- c) 질문: 앞의 스크린샷을 살펴보면 이 시스템에 몇 개의 오류가 발생했습니까?
해답: 없음
- d) 질문: 앞의 스크린샷을 살펴보면 가장 오래 실행 중인 query의 기간은 얼마입니까?
해답: 18분(답이 다를 수도 있음)

연습 10-1: 언두 데이터 관리 (계속)

- e) 더하기(+) 아이콘을 눌러 관련 그래프를 표시합니다.



- f) 질문: 몇 개의 그래프가 표시될까요?

해답: 3개 (Undo Tablespace Usage, Undo Retention Auto-Tuning 및 Undo Generation Rate)

- g) 질문: 앞의 Undo Retention Auto-Tuning 그래프를 살펴보면 이 시스템은 현재 가장 오래 실행 중인 query 시간보다 오래 Flashback을 지원할 수 있습니까?

해답: 예, 하지만 필요한 48시간을 지원하기에는 충분하지 않을 것 같습니다.

연습 10-1: 언두 데이터 관리 (계속)

- 2) 요청된 48시간 retention을 지원할 수 있도록 언두 retention 시간을 수정하고 언두 테이블스페이스 크기를 계산합니다.
- General 탭을 눌러 General Automatic Undo Management 페이지로 돌아갑니다.
 - Undo Advisor 섹션에서 "Specified manually to allow for longer duration queries or flashback"을 선택합니다.
 - Duration에 48 hours를 입력하고 Run Analysis 버튼을 누릅니다.

Undo Retention Settings Undo Retention (minutes) 15 Retention Guarantee No	Undo Tablespace for this Instance Tablespace UNDOTBS1 Change Tablespace Size (MB) 100 Auto-Extensible Yes
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 Seven Days Desired Undo Retention <input type="radio"/> Automatically chosen based on longest query in analysis period <input checked="" type="radio"/> Specified manually to allow for longer duration queries or flashback Duration 48 hours hours Run Analysis	
Analysis Results Edit Undo Tablespace Edit Undo Retention	

연습 10-1: 언두 데이터 관리 (계속)

- d) Undo Advisor가 완료되면 결과를 검토합니다.

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** [Change Tablespace](#)
Size (MB) **100**
Auto-Extensible **Yes**

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 Seven Days**
Desired Undo Retention
 Automatically chosen based on longest query in analysis period
 Specified manually to allow for longer duration queries or flashback
Duration **48** hours

Analysis Results

Selected Analysis Time Period **Jul 2, 2009 8:00:00 AM GMT+07:00 To Jul 9, 2009 8:00:00 AM GMT+07:00**
Edit Undo Tablespace Edit Undo Retention
Minimum Required Undo Tablespace Size **249** (MB)
Recommended Undo Tablespace Size (MB) **20**
TIP Recommended size is three times the minimum size to allow for workload fluctuations
Potential Problems **No Problem Found**
Recommendations **No Recommendation**

[Show Graph](#)

주: 권장되는 크기는 여기에 표시된 내용과 다를 수 있습니다.

- e) General Automatic Undo Management 페이지의 오른쪽 상단 모서리에 있는 **Show SQL** 버튼을 누릅니다.

Show SQL

[Return](#)

```
ALTER SYSTEM SET UNDO_RETENTION = 172800
```

- f) 이 명령은 48 시간 요구 사항을 지원하도록 언두 retention을 변경합니다. SQL 문을 검토하고 **Return**을 누릅니다.
- g) **Apply**를 눌러 언두 retention을 변경합니다.
- h) 이제 **Edit Undo Tablespace** 버튼을 눌러 언두 테이블스페이스 크기를 조정합니다.
- i) Datafiles로 이동하고 **Edit**를 눌러 언두 테이블스페이스에 대한 데이터 파일의 파일 크기를 변경합니다.
- j) Undo Advisor를 실행할 때 확인된 Minimum Required Undo Tablespace Size(위 스크린샷에 표시된 값인 249MB)로 파일 크기를 변경하고 **Continue**를 누릅니다.

연습 10-1: 언두 데이터 관리 (계속)

- k) Show SQL을 눌러 실행되는 SQL 명령을 확인합니다.

Show SQL

ALTER DATABASE DATAFILE '+DATA/orcl/datafile/undotbs1.258.691672083' RESIZE 249M
ALTER DATABASE DATAFILE '+DATA/orcl/datafile/undotbs1.258.691672083' AUTOEXTEND ON NEXT 5M

Return

Return을 누릅니다.

- l) Apply를 눌러 테이블스페이스 크기를 변경합니다.

- 3) Automatic Undo Management 페이지로 이동하여 바로 전 변경 사항의 결과를 확인합니다. 언두 retention 시간이 48시간 요구 사항을 지원하도록 늘어난 것을 확인할 수 있습니다. 언두 테이블스페이스 크기는 언두 테이블스페이스에 대한 데이터 파일에서 변경한 크기에 따라서도 증가합니다.

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) **2880**
Retention Guarantee **No**

Undo Tablespace for this Instance

Tablespace **UNDOTBS1** **Change Tablespace**
Size (MB) **249**
Auto-Extensible **Yes**

- a) 질문: 이 변경으로 어느 Flashback 작업이 잠재적으로 영향을 받습니까?

해답: Flashback Query, Flashback Transaction 및 Flashback Table

- b) 질문: 데이터베이스 종료 후에도 언두 데이터가 존속됩니까?

해답: 예, 언두가 지속됩니다.

단원 11의 연습

현재 상황: 방금 orcl 데이터베이스의 HR.JOBS 테이블에 의심스러운 작업이 있다는 통보를 받았습니다. 최고 급여가 이상한 방식으로 변동되는 것 같습니다. 표준 데이터베이스 감사(audit)를 활성화하고 이 테이블의 DML(데이터 조작어) 작업을 모니터하려고 합니다.

Oracle Internal & Oracle Academy
Use Only

연습 11-1: 데이터베이스 감사(Audit) 구성

DBA1 유저(oracle_4U 암호를 사용하여 SYSDBA로 연결)로 로그인하고 Enterprise Manager Database Control 또는 SQL*Plus를 통해 필요한 작업을 수행합니다. 이 연습에 대한 모든 스크립트는 /home/oracle/labs 디렉토리에 있습니다.

- Enterprise Manager를 사용하여 데이터베이스 감사(audit)를 활성화합니다.

AUDIT_TRAIL 파라미터를 **XML**로 설정합니다.

- orcl** 데이터베이스에 대해 **SYSDBA** 룰의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
- Server 탭을 누른 다음 Security 섹션에서 **Audit Settings**를 누릅니다.
- Audit Trail 섹션으로 **DB** 링크를 누릅니다.
- Initialization Parameters 페이지에서 **SPFILE** 탭을 누릅니다.
- Name 필드에 **audit**를 입력한 다음 **Go**를 누릅니다.
- audit_trail** 파라미터에 대해 **XML**을 입력합니다.
- Show SQL**을 누릅니다.

```
Show SQL
ALTER SYSTEM SET audit_trail = "XML" SCOPE=SPFILE
Return
```

- 명령문을 검토한 다음 **Return**을 누릅니다.
 - Initialization Parameters 페이지에서 **Apply**를 누릅니다.
- Static Parameter를 변경했기 때문에 데이터베이스를 재시작해야 합니다. 이렇게 하려면 **lab_11_01_02.sh** 스크립트를 실행합니다.
 - 터미널 window에서 다음을 입력합니다.

```
$ cd /home/oracle/labs
$ ./lab_11_01_02.sh
```

- 데이터베이스가 재시작된 것을 확인한 다음에는 다음 단계를 계속 수행합니다.
- Enterprise Manager로 돌아가서 **HR.JOB\$**를 Audited Objects로 선택하고 **DELETE**, **INSERT** 및 **UPDATE**를 Selected Statements로 선택합니다. 세션별로 감사(audit) 정보를 수집합니다. 데이터베이스가 재시작되었기 때문에 **DBA1** 유저로 Enterprise Manager에 다시 로그인해야 합니다.
 - Enterprise Manager window의 오른쪽 상단 모서리에서 **logout**을 누릅니다.
 - orcl** 데이터베이스에 대해 **SYSDBA** 룰의 **DBA1** 유저로 로그인합니다.
 - Database home 페이지 탭을 눌러 Enterprise Manager가 데이터베이스의 상태와 해당 에이전트 연결을 개신했는지 확인합니다.
 - Server 탭을 누른 다음 Security 섹션에서 **Audit Settings**를 누릅니다.

연습 11-1: 대비데이터베이스 감사(Audit) 구성(계속)

- e) 페이지 하단에서 **Audited Objects** 탭을 누르고 **Add** 버튼을 누릅니다.
- f) Add Audited Object 페이지에서 Object Type이 Table인지 확인하고 **Table** 필드에 **HR.JOBs**를 입력합니다(또는 손전등 아이콘을 사용하여 이 테이블을 검색).
- g) **DELETE, INSERT 및 UPDATE**를 각각 두 번 눌러 Selected Statements 영역으로 이동합니다.
- h) **Show SQL**을 누릅니다.

Show SQL	Return
AUDIT DELETE, INSERT, UPDATE ON HR.JOBs BY SESSION	

- i) 명령문을 검토한 다음 **Return**을 누릅니다.
 - j) **OK**를 눌러 이 감사를 활성화합니다.
- 4) **lab_11_01_04.sh** 스크립트를 실행하여 감사(audit)에 대한 입력을 제공합니다. 이 스크립트는 AUDIT_USER 유저를 생성하고, 이 유저로 SQL*Plus에 연결하고, MAX_SALARY 열의 값에 10을 곱합니다. 그런 다음 연결한 HR 유저가 열 값을 10으로 나눕니다. 마지막으로 AUDIT_USER 유저가 다시 삭제됩니다.
- a) 터미널 window에서 다음을 입력합니다.

```
$ cd /home/oracle/labs  
$ ./lab_11_01_04.sh
```

- 5) Enterprise Manager에서 감사(audit)된 객체를 검토합니다.
- a) **Server** 탭을 누른 다음 Security 섹션에서 **Audit Settings**를 누릅니다.
 - b) 페이지 오른쪽에 있는 Audit Trails 영역에서 **Audited Objects**를 누릅니다.
 - c) Audited Objects 페이지에서 수집된 정보를 검토합니다.

Audited Objects				
Show SQL Filter Result Return				
Schema	Object Name	User Name	Action	Time (In Session's Time Zone)
HR	JOBs	HR	SESSION REC	2009-07-09 08:57:01.812159
HR	JOBs	AUDIT_USER	SESSION REC	2009-07-09 08:57:01.71946

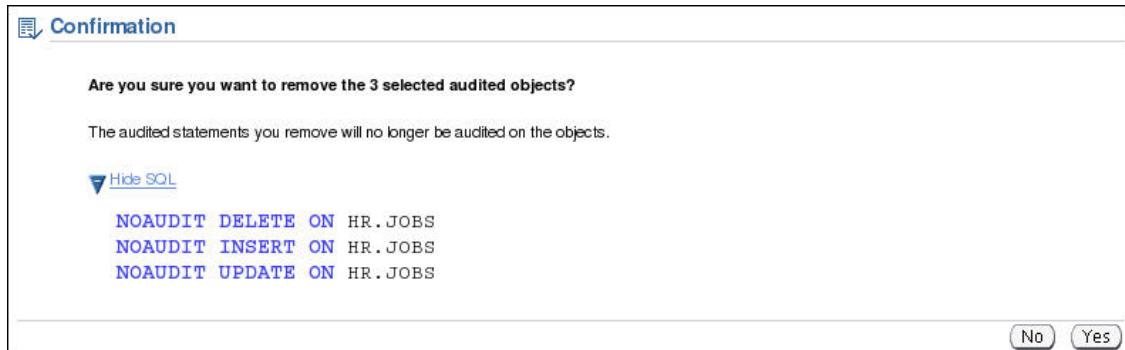
질문: 어떤 유저가 급여를 인상했고 어떤 유저가 급여를 줄였는지 알 수 있습니까?

해답: 아니오. 표준 감사(Audit) 레코드는 어떤 유저가 테이블에 액세스했는지만 표시합니다.

- d) **Return**을 누릅니다.

연습 11-1: 데이터베이스 감사(Audit) 구성(계속)

- 6) HR.JOBES에 대한 감사(audit) 설정을 연두하고 데이터베이스 감사(audit)를 비활성화한 다음 **lab_11_01_06.sh** 스크립트를 사용하여 데이터베이스를 재시작합니다.
- Audit Settings 페이지에서 페이지 하단에 있는 **Audited Objects** 탭을 누릅니다.
 - Schema에 **HR**을 입력한 다음 **Search**를 누릅니다.
 - 세 개의 행을 모두 선택한 다음 **Remove**를 누릅니다.
 - Confirmation 페이지에서 **Show SQL**을 누릅니다.



- 명령문을 검토한 다음 **Yes**를 눌러 제거를 확인합니다.
- Audit Settings 페이지의 Configuration 영역에서 **XML**을 누릅니다.
- Initialization Parameters 페이지에서 **SPFile** 탭을 누릅니다.
- SPFile 페이지에서 Name 필드에 **audit**를 입력한 다음 **Go**를 누릅니다.
- audit_trail** 파라미터에 대해 **DB**을 입력합니다.
- Show SQL**을 누릅니다.



- 명령문을 검토한 다음 **Return**을 누릅니다.
- Initialization Parameters 페이지에서 **Apply**를 누릅니다.
- Static Parameter를 변경했기 때문에 데이터베이스를 재시작해야 합니다.
이렇게 하려면 **lab_11_01_06.sh** 스크립트를 실행합니다. 터미널 window에서 다음을 입력합니다.

```
$ cd /home/oracle/labs
$ ./lab_11_01_06.sh
```

연습 11-1: 데이터베이스 감사(Audit) 구성(계속)

- 7) Audit Trail을 유지 관리합니다. 이 작업을 모두 마쳤으므로
/u01/app/oracle/admin/orcl/adump 디렉토리에서 모든 감사(Audit)
파일을 백업하고 삭제합니다.
- a) 터미널 window에서 다음을 입력합니다.

```
$ cd /u01/app/oracle/admin/orcl/adump  
$ ls
```

- b) Audit Trail 파일의 백업을 생성한 다음 파일을 제거합니다.

```
$ tar -czf $HOME/audit_today.tar.z *  
$ rm -f *
```

- c) 터미널 window를 닫습니다.

단원 12의 연습

현재 상황: 유저에게 영향을 미치기 전에 일반적인 문제를 해결할 수 있도록 orcl 데이터베이스를 사전 모니터(Proactive Monitoring)하려고 합니다. 이 연습 세션에서는 제공되는 도구에 익숙해질 수 있도록 인위적으로 몇 가지 문제를 발생시킵니다. 먼저 이 연습에 필요한 데이터베이스 환경을 설정하는 스크립트를 실행합니다.

Oracle Internal & Oracle Academy
Use Only

연습 12-1: 대/이터페이스 유지 관리

- 1) TBSSPC라는 새 로컬관리방식 테이블스페이스를 생성합니다. 이 테이블스페이스는 +DATA 디스크 그룹에 크기가 50MB인 데이터 파일을 포함합니다. TBSSPC 테이블스페이스가 ASSM(Automatic Segment Space Management)을 사용하지 않는지 확인합니다. **lab_12_01_01.sh** 스크립트를 실행하여 이러한 작업을 수행합니다. 터미널 window에서 다음을 입력합니다.

```
$ cd /home/oracle/labs
$ cat lab_12_01_01.sh
...
sqlplus / as sysdba << END

set echo on

drop tablespace TBSSPC including contents and datafiles;

CREATE SMALLFILE TABLESPACE "TBSSPC"
DATAFILE '+DATA' SIZE 50M
AUTOEXTEND ON NEXT 10M MAXSIZE 200M
LOGGING
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT MANUAL;

exit;
END

$ ./lab_12_01_01.sh
```

- 2) 새 SPCT 유저를 암호 oracle_4U를 사용하여 생성합니다. TBSSPC 테이블스페이스를 기본 테이블스페이스로 지정합니다. TEMP 테이블스페이스를 임시 테이블스페이스로 지정합니다. SPCT 유저에게 CONNECT, RESOURCE 및 DBA 를을 부여합니다. **lab_12_01_02.sh** 스크립트를 실행하여 이러한 작업을 수행합니다. 터미널 window에서 다음을 입력합니다.

```
$ cat lab_12_01_02.sh
...
sqlplus / as sysdba << END

set echo on

drop user spct cascade;

create user spct identified by oracle_4U
default tablespace TBSSPC
temporary tablespace temp;

grant connect, resource, dba to spct;

exit;
END

$ ./lab_12_01_02.sh
```

연습 12-1: 데이터베이스 유지 관리 (계속)

- 3) DBMS_ADVISOR 패키지를 사용하여 데이터베이스 작업 시간을 30분으로 설정합니다. SPCT 유저로 SPCT 테이블을 삭제했다가 생성한 다음 이 테이블에 대한 통계를 수집합니다. AWR(Automatic Workload Repository)에서 스냅샷을 생성합니다. **lab_12_01_03.sh** 스크립트를 실행하여 이러한 작업을 수행합니다. 터미널 window에서 다음을 입력합니다.

```
$ cat lab_12_01_03.sh
...
sqlplus / as sysdba << EOF

set echo on

exec
dbms_advisor.set_default_task_parameter('ADDM', 'DB_ACTIVITY_MIN', 30);

connect spct/oracle_4U

drop table spct purge;
create table spct(id number, name varchar2(2000));

exec DBMS_STATS.GATHER_TABLE_STATS(
ownname=>'SPCT', tabname=>'SPCT',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE);

exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT();

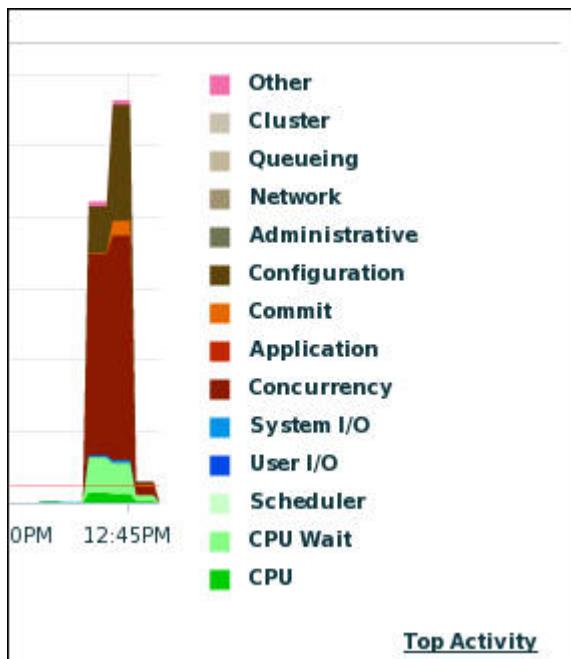
exit;
EOF
$ ./lab_12_01_03.sh
```

- 4) 분석할 작업을 생성합니다. **lab_12_01_04.sh** 스크립트를 실행하여 이러한 작업을 수행합니다. 터미널 window에서 다음을 입력합니다. 여러 PL/SQL 프로시저가 완료된 것을 확인한 후 다시 명령 프롬프트를 표시하기 위해 [Enter]를 눌러야 할 수도 있습니다.

```
$ ./lab_12_01_04.sh
```

연습 12-1: 데이터베이스 유지 관리 (계속)

- 5) Enterprise Manager에서 SYSDBA로 연결된 유저를 통해 Performance 페이지를 검토합니다. 15초 Refresh 주기로 성능 데이터를 실시간으로 확인합니다. 잠시 후 Average Active Sessions 그래프에 스파이크가 표시됩니다. 이것이 분석할 작업입니다. 그래프를 보면 이 Instance에 동시성 문제가 있음을 사전에 파악할 수 있습니다.
- orcl 데이터베이스에 대해 **SYSDBA** 를의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
 - Performance 탭을 누릅니다.



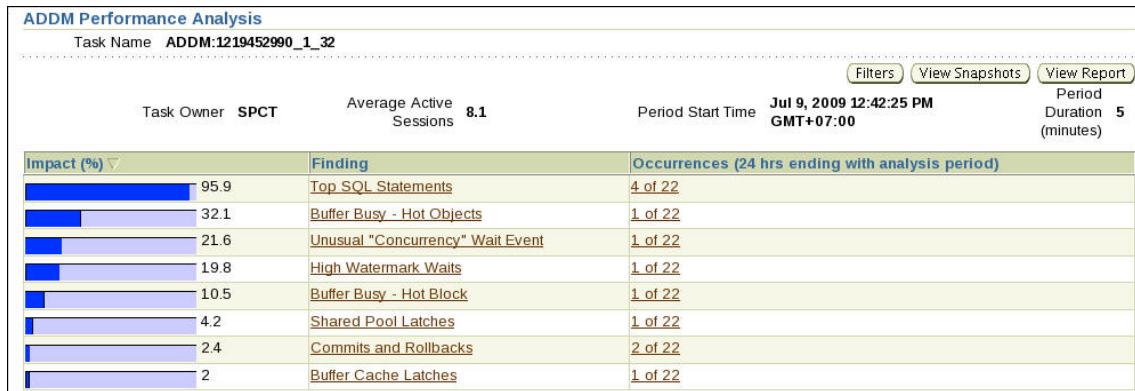
주: 작업 로드를 실행하는 시간에 따라 예상 해답으로 제공된 그래프와 다르게 나타날 수 있습니다.

- 스파이크가 완료되면 **lab_12_01_05.sh** 스크립트를 실행합니다. 이 스크립트는 새 스냅샷을 생성하고 SPCT 테이블에 대한 통계를 수집합니다.
주: 모든 환경에서 동일 성능 문제를 일으키기란 쉽지 않은 일입니다. 테스트를 보다 성공적으로 수행하기 위해서는 추가로 몇 분 정도 기다려서 스파이크가 완전히 종료된 다음에 스크립트를 실행하십시오. 스파이크가 완료되면 터미널 window에서 다음을 입력합니다.

```
$ ./lab_12_01_05.sh
```

연습 12-1: 데이터베이스 유지 관리 (계속)

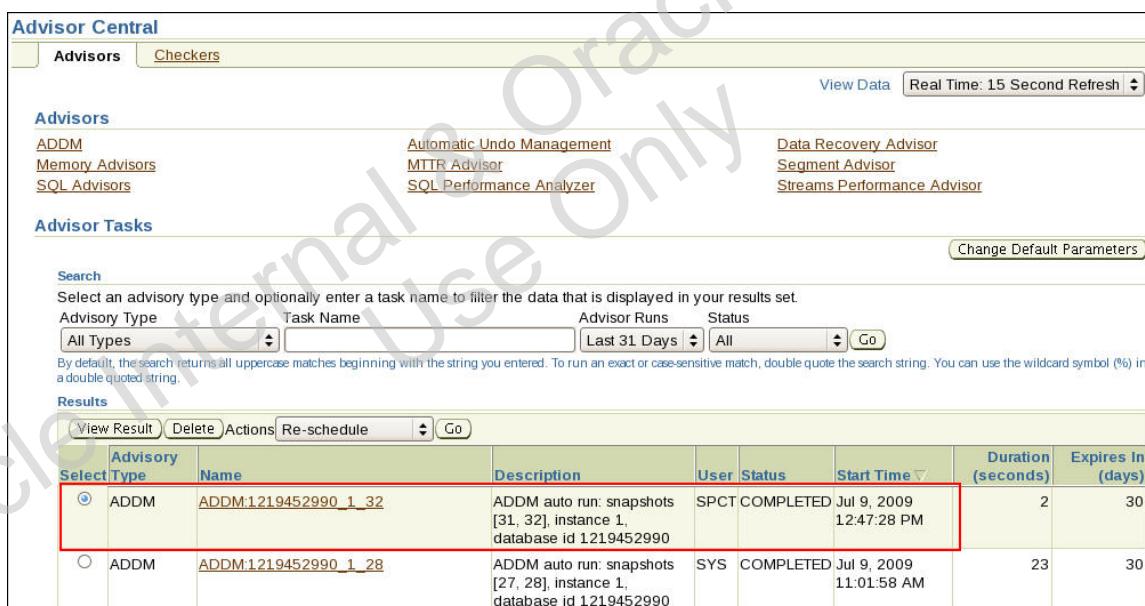
- 6) 영향을 미치는 순서대로 **Performance Analysis** 결과를 살펴봅니다. 여러 경로를 통해 이 정보에 액세스할 수 있습니다. 결과는 다음과 같아야 합니다.



Performance Analysis 섹션을 살펴보면 첫번째 결과가 시스템에 미치는 영향의 비율이 높다는 것을(이 예제에서는 95.9%) 알 수 있습니다. 따라서 첫번째로 할 일은 이 결과를 더욱 자세히 검토하는 것입니다. 그러나 이 SQL 문을 살펴보는 것은 아직 데이터베이스의 동시성 문제를 파악하는 데 도움이 되지 않습니다.

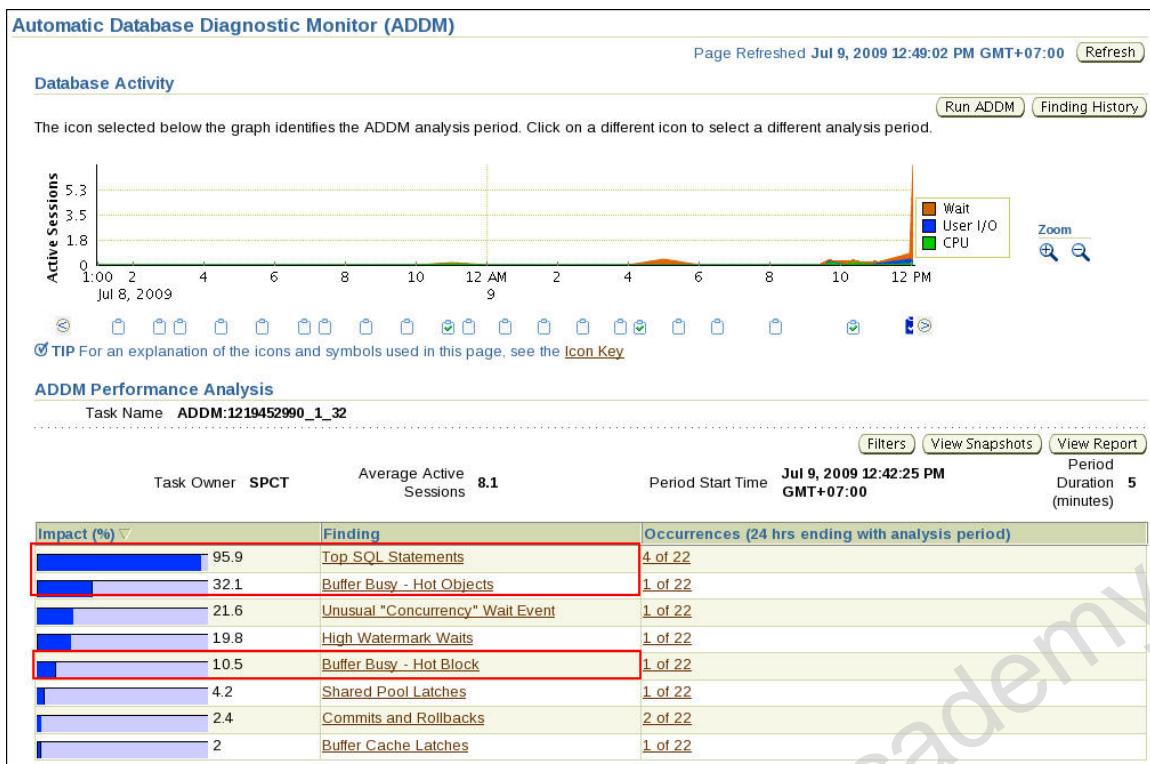
심각도 순서에 따라 다른 결과들도 조사합니다. 특히 Buffer Busy 결과에 주의합니다. Buffer Busy 결과 중 하나에서는 SPCT 테이블에서 읽기/쓰기 경합이 있음을 알 수 있습니다. 권장되는 작업은 SPCT 테이블에 ASSM(Automatic Segment Space Management) 기능을 사용하는 것입니다. 다음 단계에서는 이 연습을 수행합니다.

- Database home 페이지로 이동한 다음 페이지 아래쪽에서 **Advisor Central**을 누릅니다.
- ADDM 작업이 이미 표시되어 있어야 합니다. 그렇지 않은 경우 해당 작업을 검색하여 이 페이지에 표시합니다.



연습 12-1: 데이터베이스 유지 관리 (계속)

- c) 작업을 선택한 다음 **View Result** 버튼을 누르거나 작업 이름을 누릅니다.
- d) ADDM 페이지가 표시되어 ADDM 실행에 따른 자세한 결과를 보여 줍니다.
 주: lab_12_01_05.sh 스크립트를 실행했을 때 ADDM 성능 분석이 이미 수행되었으므로 Run ADDM Now 버튼을 누르지 마십시오. 지금 버튼을 누르면 빈 결과 집합이 생성됩니다.



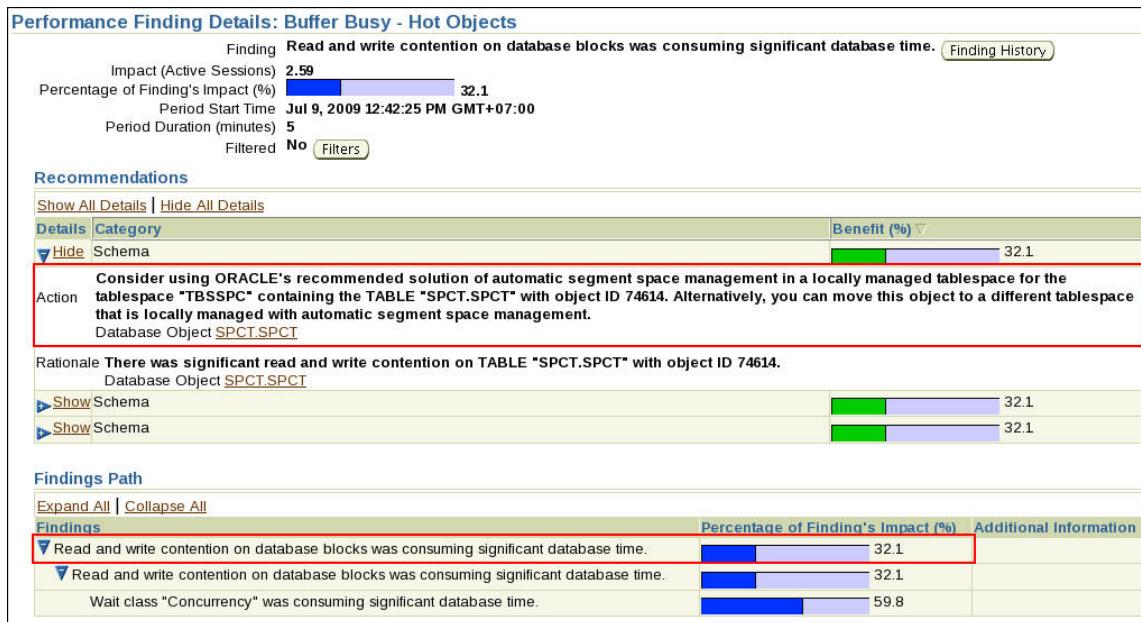
Performance Analysis 섹션을 살펴보면 첫번째 결과가 시스템에 미치는 영향이 높다는 것을(이 예제에서는 95.9%) 알 수 있습니다. 또한 Buffer Busy 결과도 확인할 수 있습니다. Top SQL 문 결과는 이렇게 큰 비율로 시스템에 영향을 주기 때문에 먼저 해야 할 일은 이 결과를 보다 자세하게 확인하는 것입니다.

주: 강의실 시스템에는 성능에 영향을 줄 수 있는 여러 요인들이 있기 때문에 결과가 정확하게 표시되지 않을 수 있습니다. 결과는 다른 순서로 나타날 수 있습니다. 이전 스크린샷에 표시된 것과 비슷한 결과가 표시되지 않으면 이 연습을 재시작해야 할 수 있습니다. 예상한 결과가 계속 표시되지 않으면 lab_12_01_04.sh 및 lab_12_01_04.sql 스크립트를 수정하여 로드를 조정해야 할 수 있습니다. 이러한 경우에는 강사에게 도움을 요청하십시오. 로드를 너무 늘리지 않도록 주의하십시오. 그렇지 않으면 시스템 속도가 너무 느려질 수 있습니다.

- e) Finding 열에서 "Top SQL Statements" 링크를 누릅니다.
- f) Performance Finding Details 페이지에서 권장 사항을 검토합니다. 그러나 이 SQL 문을 살펴보는 것은 아직 데이터베이스의 동시성 문제를 파악하는 데 도움이 되지 않습니다. 웹 브라우저에서 **Back** 아이콘을 누릅니다.

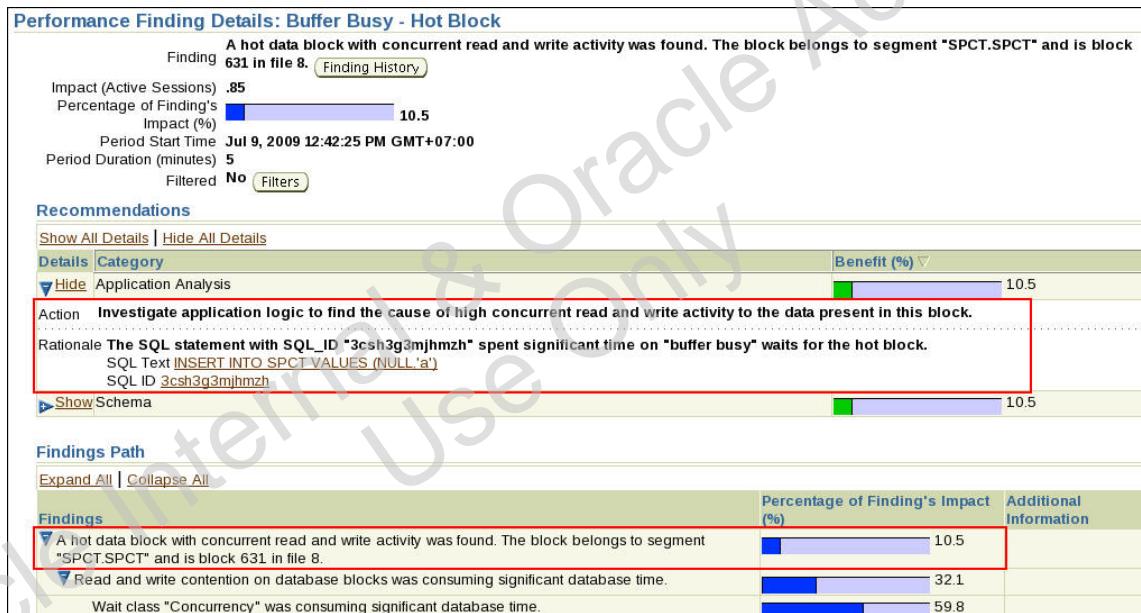
연습 12-1: 데이터베이스 유지 관리 (계속)

- g) Automatic Database Diagnostic Monitor (ADDM) 페이지에서 Buffer Busy 결과를 확인합니다. Buffer Busy 결과 중 첫번째 항목(이 경우에는 Buffer Busy – Hot Objects 링크)을 누릅니다.



이 결과에서는 데이터베이스 블록에서 읽기/쓰기 경합이 있는 것을 알 수 있습니다. 권장 방법은 Automatic Segment Space Management에서 로컬관리방식으로 관리되는 테이블스페이스를 사용하는 것입니다.

- h) ADDM 페이지로 돌아가서 다른 Buffer Busy 결과를 확인합니다. 이러한 결과 중 하나는 다음과 비슷하게 표시됩니다.



이 결과에서는 SPCT.SPCT 테이블에 속하는 핫 데이터 블록이 있음을 알 수 있습니다. 권장 방법은 응용 프로그램 논리에서 원인을 조사하는 것입니다.

연습 12-1: 데이터베이스 유지 관리 (계속)

- 7) 권장 방법에 따라 Automated Segment Space Management를 사용합니다. 이렇게 하려면 객체를 재생성해야 합니다. +DATA 디스크 그룹에서 50MB의 데이터 파일이 들어 있는 TBSSPC2라는 새 로컬관리방식 테이블스페이스를 생성합니다. TBSSPC2 테이블스페이스가 Automatic Segment Space Management 기능을 사용하는지 확인합니다. 그런 다음 **lab_12_01_07.sh** 스크립트를 실행하여 SPCT 테이블을 삭제했다가 새 테이블스페이스에서 재생성한 다음 통계를 수집하고 새 스냅샷을 생성합니다.
- Enterprise Manager에서 **Server** 탭을 누른 다음 Storage 섹션에서 **Tablespaces**를 누릅니다.
 - Create**를 누릅니다.
 - 테이블스페이스 이름으로 TBSSPC2를 입력하고 Extent Management가 **Locally Managed**, Type이 **Permanent**, Status가 **Read Write**이고 Use bigfile tablespace가 선택되지 않았는지 확인합니다.
 - Datafiles** 영역에서 **Add**를 누릅니다.
 - Add Datafile 페이지에서 DiskGroup이 **DATA**인지 확인하고 File Size로 **50 MB**를 입력합니다.
 - Continue**를 누릅니다.
 - Storage 탭을 누르고 Extent Allocation 및 Segment Space Management가 **Automatic**이고 Logging이 활성화되었는지 확인합니다.
 - General** 탭을 누릅니다.
 - Show SQL을 누르고 실행할 SQL을 확인한 다음 **Return**을 누릅니다.

```
Show SQL
CREATE SMALLFILE TABLESPACE "TBSSPC2" DATAFILE '+DATA' SIZE 50M LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
Return
```

j) OK를 누릅니다. Confirmation 성공 확인 메시지가 나타납니다.

k) 터미널 window에서 다음을 입력합니다.

```
$ ./lab_12_01_07.sh
```

- 8) 작업 로드를 다시 실행합니다. **lab_12_01_04.sh** 스크립트를 사용합니다.

터미널 window에서 다음을 입력합니다. 여러 PL/SQL 프로시저가 완료된 것을 확인한 후 다시 명령 프롬프트를 표시하기 위해 [Enter]를 눌러야 할 수도 있습니다.

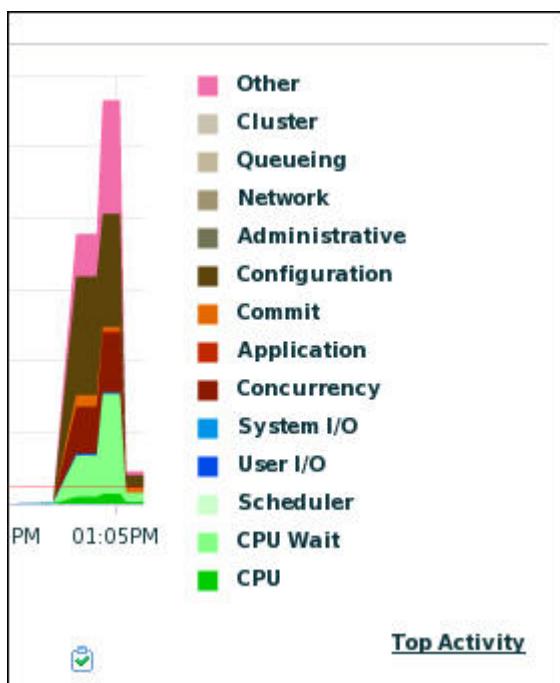
```
$ ./lab_12_01_04.sh
```

연습 12-1: 데이터베이스 유지 관리 (계속)

- 9) Enterprise Manager에서 SYSDBA로 연결된 유저를 통해 Performance 페이지를 검토합니다. 15초 Refresh 주기로 성능 데이터를 실시간으로 확인합니다. 잠시 후 Average Active Sessions 그래프에 스파이크가 표시됩니다.

스파이크가 완료되면 **lab_12_01_05.sh** 스크립트를 다시 실행합니다. 이 스크립트는 새 스냅샷을 생성하고 ADDM 테이블에 대한 통계를 수집합니다.

- orcl** 데이터베이스에 대해 **SYSDBA** 툴의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
- Performance** 탭 페이지를 누릅니다. Active Sessions 차트에서 스파이크가 완료되는 것을 확인합니다.



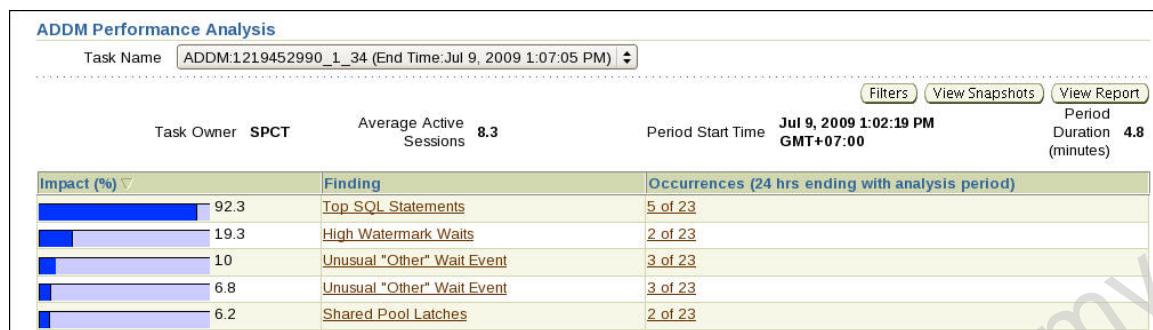
- 스파이크가 종료되면 **lab_12_01_05.sh** 스크립트를 실행하여 새 스냅샷 생성을 강제 시행하고 SPCT 테이블에 대한 통계를 수집합니다. 터미널 window에서 다음을 입력합니다.

```
$ ./lab_12_01_05.sh
```

연습 12-1: 데이터베이스 유지 관리 (계속)

10) Advisor Central 링크에서 ADDM을 검토합니다.

- a) Database home 페이지로 이동한 다음 페이지 아래쪽에서 **Advisor Central**을 누릅니다.
- b) 맨 위의 ADDM 작업 이름을 누릅니다.
- c) 읽기/쓰기 경합에 대한 Buffer Busy 결과가 더 이상 없음을 확인할 수 있습니다. Automatic Autoextend Segment 기능을 사용하는 로컬관리방식의 TBSSPC2 테이블스페이스로 ADDM 테이블을 이동하여 이 경합 문제의 근본 원인을 확실히 해결했습니다.
주: 추가적인 Buffer Busy 결과(영향 비율이 낮은) 및 성능 향상을 위한 추가 권장 사항이 표시될 수도 있지만 이 과정에서는 이러한 문제의 해결에 대해서는 다루지 않습니다.



11) 다른 연습 세션에 영향을 미치지 않도록 **lab_12_01_11.sh** 스크립트를 실행하여 작업 환경을 정리합니다.
터미널 window에서 다음을 입력합니다.

```
$ ./lab_12_01_11.sh
```

단원 13의 연습

현재 상황: 유저가 인적 자원 관리(HR) 및 주문 입력 응용 프로그램과 관련된 작업 성능이 보통 이하라는 사실에 불만을 나타내고 있습니다. 다른 DBA 직원에게 문의해 본 결과 최근에 HR 스키마에 속하는 일부 테이블에 대해 유지 관리가 수행되었다는 것을 알게 되었습니다. 적절한 변경 내용을 적용하여 성능 문제를 해결합니다. SQL 스크립트 파일은 /home/oracle/labs 디렉토리에 제공되어 있습니다. 다른 디렉토리는 개별적으로 이름이 지정됩니다.

Oracle Internal & Oracle Academy
Use Only

연습 13-1: 성능 관리

- 1) SQL*Plus에 SYS 유저로 로그인한 다음 **lab_13_01_01.sql** 스크립트를 실행하여 HR 스키마의 테이블에 대한 유지 관리를 수행합니다.

```
$ cd ~/labs
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba
SQL> @lab_13_01_01.sql
```

- 2) HR 응용 프로그램 유저로부터 특정 query가 정상적인 실행 시간보다 오래 걸린다는 전화를 받습니다. 이 query는 **lab_13_01_02.sql** 스크립트에 있습니다. 이 파일을 실행하려면 SQL*Plus에서 다음을 입력합니다.

```
SQL> CONNECT hr
Password: oracle_4U    <<<Password does not appear on screen
Connected.
SQL> @lab_13_01_02.sql
```

- 3) Enterprise Manager를 사용하여 위의 명령문이 방금 실행된 HR 세션을 찾아 해당 명령문의 실행 계획을 확인합니다.
- Enterprise Manager에서 **Performance** 탭을 누르고 Additional Monitoring Links 섹션에서 **Search Sessions**를 누릅니다.
 - Search Sessions 페이지에서 검색 조건을 "DB User"로 변경하고 오른쪽 필드에 **HR**을 입력한 다음 **Go**를 누릅니다.
 - Results list에서 **SID** 번호를 누릅니다.
 - 이 세션에 대한 Session Details 페이지가 나타납니다. Application 섹션에서 Previous SQL 레이블 오른쪽에 있는 해시 값 링크를 누릅니다.

연습 13-1: 성능 관리 (계속)

- e) SQL Details 페이지에 문제가 되는 해당 세션에 의해 실행된 마지막 SQL 문에 대한 세부 정보가 나타납니다. Plan 탭을 눌러 query에 대한 실행 계획을 봅니다.

The screenshot shows the 'SQL Details' page for session ID bckcqw5pd108f. The query is:

```
select *
from hr.employees
where employee_id = 200
```

The 'Plan' tab is selected. The execution plan table has the following columns: Operation, Object, Order, Rows, Bytes, Cost, CPU (%), Time, Query Block Name/Object Alias, Filter, and Projection. One row in the table is highlighted with a red box, showing the operation as 'TABLE ACCESS FULL' for the object 'EMPLOYEES'. The 'Projection' column shows the query as "EMPLOYEE_ID=200 [NUMBER.22], *EMP...".

- f) Operation 열에서 이 query가 전체 테이블 스캔(TABLE ACCESS FULL)을 수행하고 있음을 확인할 수 있습니다. query의 조건이 primary key(EMPLOYEE_ID)에서 동등 비교라는 사실을 알고 있기 때문에 primary key 인덱스의 상태를 조사하려고 합니다.
- 4) Enterprise Manager를 사용하여 EMPLOYEE 테이블의 EMPLOYEE_ID에 대한 인덱스의 상태를 검사합니다. 상태가 VALID인지 확인합니다.
- Database Home 페이지에서 Schema 탭과 Indexes를 차례로 누릅니다.
 - Search By 칸으로 Table Name을 선택합니다.
 - Schema 필드에 HR을 입력합니다.
 - Object Name에 EMPLOYEES를 입력합니다.
 - Go를 누르면 여섯 개의 인덱스 리스트가 나타납니다.

연습 13-1: 성능 관리 (계속)

- f) 이름이 **EMP_EMP_ID_PK**인 인덱스를 누릅니다.

Indexes

Object Type: Index

Search

Enter a schema name and an object name to filter the data that is displayed in your results set.

Search By: Table Name

Schema: HR

Object Name: EMPLOYEES

Go

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.

Selection Mode: Single

Create

Edit View Delete Actions Create Like Go

Select	Table Owner	Table	Indexed Columns	Index Owner	Index	Table Type	Tablespace	Partitioned	Last Analyzed
C	HR	EMPLOYEES	EMAIL	HR	EMP_EMAIL_UK	TABLE	EXAMPLE	NO	Aug 26, 2008 12:46:38 AM GMT+07:00
C	HR	EMPLOYEES	MANAGER_ID, LAST_NAME	HR	EMP_NAME_IX	TABLE	EXAMPLE	NO	Aug 26, 2008 12:46:38 AM GMT+07:00
C	HR	EMPLOYEES	MANAGER_ID	HR	EMP_MANAGER_IX	TABLE	EXAMPLE	NO	Aug 26, 2008 12:46:38 AM GMT+07:00
C	HR	EMPLOYEES	DEPARTMENT_ID	HR	EMP_DEPARTMENT_IX	TABLE	EXAMPLE	NO	Aug 26, 2008 12:46:38 AM GMT+07:00
C	HR	EMPLOYEES	JOB_ID	HR	EMP_JOB_IX	TABLE	EXAMPLE	NO	Aug 26, 2008 12:46:38 AM GMT+07:00
C	HR	EMPLOYEES	EMPLOYEE_ID	HR	EMP_EMP_ID_PK	TABLE	EXAMPLE	NO	Aug 26, 2008 12:46:38 AM GMT+07:00

- g) View Index 페이지에서 인덱스 상태가 UNUSABLE임을 알 수 있습니다.

View Index: HR.EMP_EMP_ID_PK

Actions: Create Like, Go, Edit, OK

General

Name: **EMP_EMP_ID_PK**
Schema: **HR**
Tablespace: **EXAMPLE**
Index Type: **Normal**
Status: **UNUSABLE**

연습 13-1: 성능 관리 (계속)

- 5) 한 인덱스가 비VALID 상태로 확인되어 모든 인덱스를 검사하려고 합니다. HR 유저로 SQL*Plus를 사용하여 STATUS가 VALID가 아닌 **HR** 스키마 인덱스를 찾습니다. 이렇게 하려면 STATUS 열에 조건을 사용하여 데이터 딕셔너리 뷰를 query하면 됩니다.
- a) 아직 HR 유저로 로그인하고 있는 SQL*Plus 세션으로 이동하여 다음 query를 실행합니다.

```
SQL> select index_name, table_name, status
      from user_indexes where status <> 'VALID';

INDEX_NAME          TABLE_NAME        STATUS
-----              -----
EMP_EMAIL_UK        EMPLOYEES        UNUSABLE
EMP_EMP_ID_PK       EMPLOYEES        UNUSABLE
EMP_DEPARTMENT_IX   EMPLOYEES        UNUSABLE
EMP_JOB_IX          EMPLOYEES        UNUSABLE
EMP_MANAGER_IX      EMPLOYEES        UNUSABLE
EMP_NAME_IX         EMPLOYEES        UNUSABLE

6 rows selected.

SQL>
```

- b) 출력에서 **EMPLOYEES** 테이블에 모두 여섯 개의 인덱스가 나열되는지 확인합니다. 이 문제를 이제 해결해야 합니다.
- 6) 여기에서는 Enterprise Manager를 사용하여 UNUSABLE로 표시된 HR 스키마에 있는 모든 인덱스를 재구성합니다.
- Enterprise Manager에서 EMP_EMP_ID_PK 인덱스가 표시된 페이지의 Actions list에서 **Reorganize**를 선택한 다음 **Go**를 누릅니다.
 - Reorganize Objects 페이지에서 **Add**를 눌러 다른 다섯 개의 인덱스를 각각 재구성 작업에 추가합니다.
 - Add 화면의 Type drop-down list에서 **Indexes**를 선택하고 Schema 필드에 **HR**을 입력합니다. **Search**를 누릅니다.

연습 13-1: 성능 관리 (계속)

- d) 이름이 "EMP_"로 시작하는 다섯 개의 다른 인덱스를 선택합니다.

Objects: Add

Database Instance orcl.oracle.com
Logged In As SYS

Search

Type: Indexes
Schema: HR
Object Name:
Partition Name:
Tablespace:
Search

Include minimum size filter in search query
Minimum Size (KB):

Available Objects: Indexes

Select	Name	Type	Tablespace	Size (KB)	Partitioned	IOT
<input type="checkbox"/>	HR.DEPT_ID_PK	Index	EXAMPLE	64	NO	N/A
<input type="checkbox"/>	HR.DEPT_LOCATION_IX	Index	EXAMPLE	64	NO	N/A
<input checked="" type="checkbox"/>	HR.EMP_DEPARTMENT_IX	Index	EXAMPLE	64	NO	N/A
<input checked="" type="checkbox"/>	HR.EMP_EMAIL_UK	Index	EXAMPLE	64	NO	N/A
<input checked="" type="checkbox"/>	HR.EMP_JOB_IX	Index	EXAMPLE	64	NO	N/A
<input checked="" type="checkbox"/>	HR.EMP_MANAGER_IX	Index	EXAMPLE	64	NO	N/A
<input checked="" type="checkbox"/>	HR.EMP_NAME_IX	Index	EXAMPLE	64	NO	N/A
<input type="checkbox"/>	HR.JHIST_DEPARTMENT_IX	Index	EXAMPLE	64	NO	N/A
<input type="checkbox"/>	HR.JHIST_EMPLOYEE_IX	Index	EXAMPLE	64	NO	N/A
<input type="checkbox"/>	HR.JHIST_EMP_ID_ST_DATE_PK	Index	EXAMPLE	64	NO	N/A

Previous 1-10 of 17 Next 7

- e) OK를 눌러 Reorganize Objects: Objects 페이지로 돌아갑니다.
- f) EMPLOYEES 테이블에 대한 6개 인덱스가 모두 나열되었는지 확인하고 Next를 누릅니다.
- g) Options의 기본 설정을 그대로 유지하고 Next를 누릅니다. 재구성 스크립트 생성이 발생한 다음 Impact Report가 나타납니다.
- h) Impact Report에 보고된 문제가 없는지 확인한 다음 Next를 누릅니다.
- i) Schedule 페이지에서 Host Credentials 아래의 Username 및 Password에 각각 oracle을 입력합니다.
- j) Next를 누릅니다.
- k) Review 페이지에서 Submit Job을 누릅니다.
- l) Confirmation 페이지가 나타나면 View job Details를 눌러 작업 상태를 확인합니다.

Database Instance: orcl.oracle.com >

Confirmation

The reorganization job named REORGANIZE_ORCL.ORACLE.COM_1 has been submitted.

View Job Details

연습 13-1: 성능 관리 (계속)

- m) 작업이 성공한 것으로 나타날 때까지 브라우저에서 **Reload**를 누릅니다.

Job Run: REORGANIZE_ORCL.ORACLE.COM_1	
Page Refreshed Jul 9, 2009 10:53:29 AM GMT+07:00	
Delete Run Edit View Definition	
Summary	
Status	Succeeded
Scheduled	Jul 9, 2009 10:52:46 AM (UTC+07:00)
Started	Jul 9, 2009 10:52:46 AM (UTC+07:00)
Ended	Jul 9, 2009 10:52:57 AM (UTC+07:00)
Elapsed Time	11 seconds
Notification	No
Type	Reorganize
Owner	SYS
Description	
DB Username	SYS
Username	oracle
Script	/u01/app/oracle/product/11.2.0/d...

- 7) HR 유저가 로그인한 SQL*Plus 세션으로 돌아가서 **lab_13_01_07.sql**

스크립트를 실행하여 동일한 유형의 query를 실행합니다. 그런 다음 이 세션에서 실행된 마지막 SQL 문의 계획을 확인하는 단계를 반복하여 계획이 변경되었는지 확인합니다.

- a) SQL*Plus 프롬프트에서 다음을 입력합니다.

```
SQL> @lab_13_01_07.sql
```

- b) 3단계에 나열된 작업을 반복하여 query에 대한 실행 계획을 확인합니다. 이제 아이콘에 인덱스 사용이 표시됩니다. **View Table**을 누릅니다. 이제 계획이 인덱스 고유 스캔을 사용합니다.

Details								
Select the plan hash value to see the details below. Plan Hash Value 1833546154								
Statistics	Activity	Plan	Plan Control	Tuning History	SQL Monitoring			
Data Source	CURSOR Cache	Capture Time	Jul 9, 2009 10:55:37 AM (UTC+07:00)			Parsing Schema	HR	Optimizer Mode ALL_ROWS
Additional Information								
Expand All Collapse All								
Operation	Object	Order	Rows	Bytes	Cost	CPU (%)	Query Block Time	Name/Object Alias
▼ SELECT STATEMENT		3			1	100		
▼ TABLE ACCESS BY INDEX ROWID	EMPLOYEES	2	1	69	1	0	0:0:1 SEL\$1 / EMPLOYEES@SEL\$1	"EMPLOYEE_ID"[NUMBER,22], "EMP...]
INDEX UNIQUE SCAN	EMP_EMP_ID_PK	1			0		SEL\$1 / EMPLOYEES@SEL\$1	"EMPLOYEE_ID"=201 "EMPLOYEES".ROWID[ROWID,10], ...]

- c) SQL*Plus 세션을 종료합니다.

- 8) 실행 계획에 어떤 차이가 있으며 그 이유는 무엇입니까?

해답: 인덱스 재구성 후에 인덱스를 사용할 수 있기 때문에 명령문 실행에는 전체 테이블 스캔 대신 고유 인덱스 스캔이 사용됩니다.

연습 13-1: 성능 관리 (계속)

- 9) SYS 유저로 **lab_13_01_09.sql** 스크립트를 실행하여 instance에서 작업 로드를 시뮬레이트합니다. 보고되는 SID 값을 기록해 두십시오.

보고된 SID 값: _____

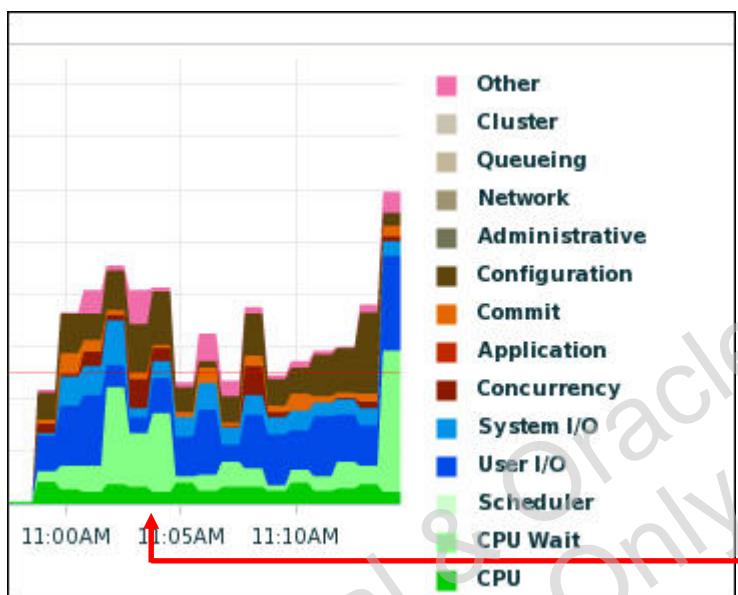
이 스크립트는 완료하는 데 20분 정도 걸립니다. 따라서 별도의 터미널 window에서 실행하고 그 동안 이 연습 과정을 계속 진행하십시오.

주의: 이 스크립트는 CPU 및 디스크 I/O에 과도한 로드를 발생시키므로 Database Control에 대한 응답 시간이 느려집니다.

```
$ sqlplus / as sysdba  
SQL> @lab_13_01_09.sql
```

- 10) Enterprise Manager로 돌아가서 데이터베이스 성능을 검사합니다.

- Enterprise Manager에서 Performance 페이지로 이동하여 시스템 성능을 조사합니다.
- 로드 생성 스크립트의 영향이 그래프에 나타나는 것을 확인하려면 1-2분 정도 기다려야 하는 경우도 있습니다.



계속하기 전에 Average
Active Sessions 그래프에서
스파이크가 시작될 때까지
기다립니다.

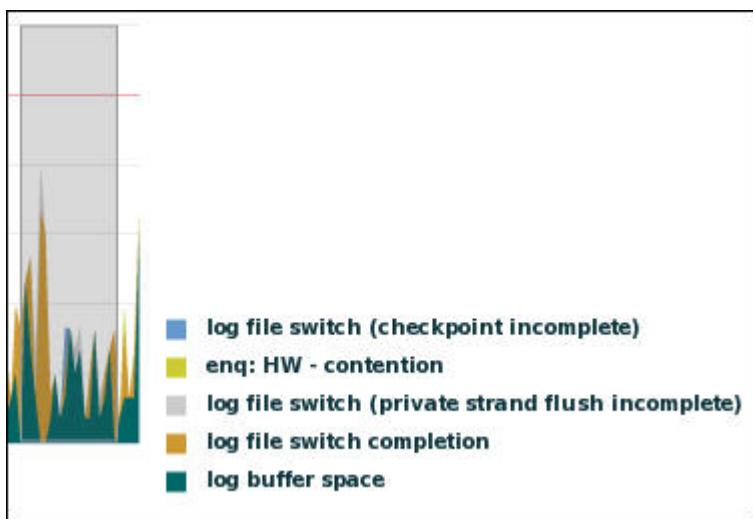
연습 13-1: 성능 관리 (계속)

질문 1: Average Active Sessions 그래프에서 활성 세션이 기다리는 두 가지 주요 범주는 무엇입니까?

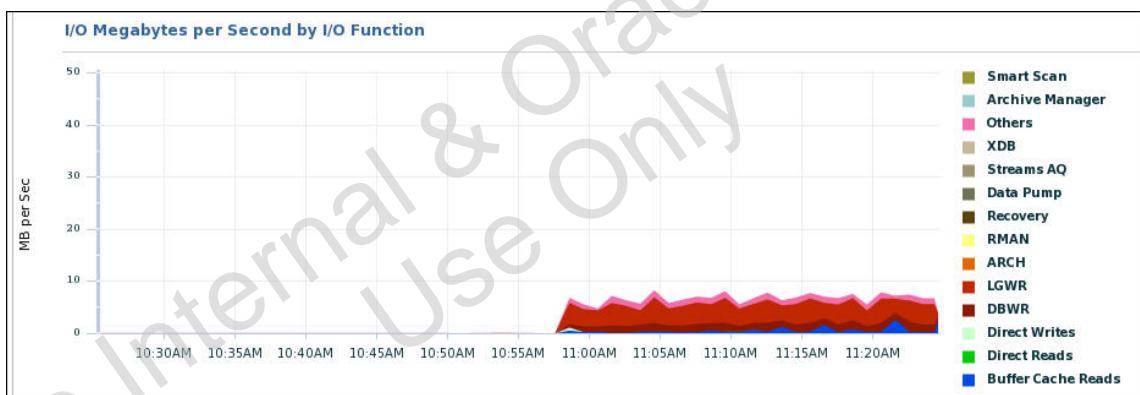
해답: 이 예제에서는 CPU Wait와 User I/O가 상당히 높은 것으로 보입니다. Configuration도 대기 활동이 높습니다. 결과 값은 여기에 나오는 것과 다를 수 있습니다.

질문 2: Configuration 대기 범주에서 대기 시간에 기여한 제공자 중 하나는 무엇입니까? Configuration을 눌러 그래프를 확인합니다.

해답: 모든 제공자가 대기 시간에 기여했지만 log file switch completion 및 log buffer space가 가장 대기 시간이 긴 것으로 보입니다.



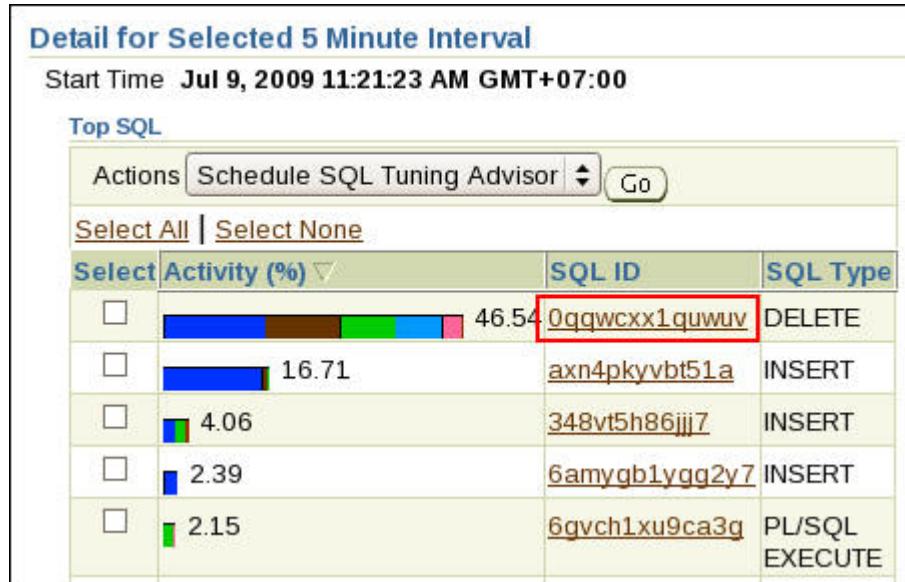
질문 3: Back을 누른 다음 Performance 페이지에서 Settings를 누릅니다. Detail Chart Settings에서 Default View에 I/O를 선택하고 I/O Chart Settings에 I/O Function을 선택한 다음 OK를 누릅니다. IO 차트로 이동하여 디스크 쓰기가 가장 많은 프로세스를 확인합니다.



해답: LGWR

연습 13-1: 성능 관리 (계속)

- c) Additional Monitoring Links 영역에서 **Top Activity**를 누릅니다.
- d) Top SQL 영역에 나열된 첫번째 SQL 문의 **SQL ID**를 누릅니다.



- e) 첫번째 SQL 문을 확인합니다.
- 11) 로드를 생성하는 세션을 종료합니다. 단계 9에서 기록한 세션 ID를 사용합니다.
세션 ID는 Detail for Selected 5 Minute Interval의 **SID** 열에 나열됩니다.
- a) 이전에 기록한 세션 ID의 **SID** 번호를 누릅니다. 이 번호는 **Detail for Selected 5 Minute Interval** 머리글 아래 있습니다.

Detail for Selected 5 Minute Interval

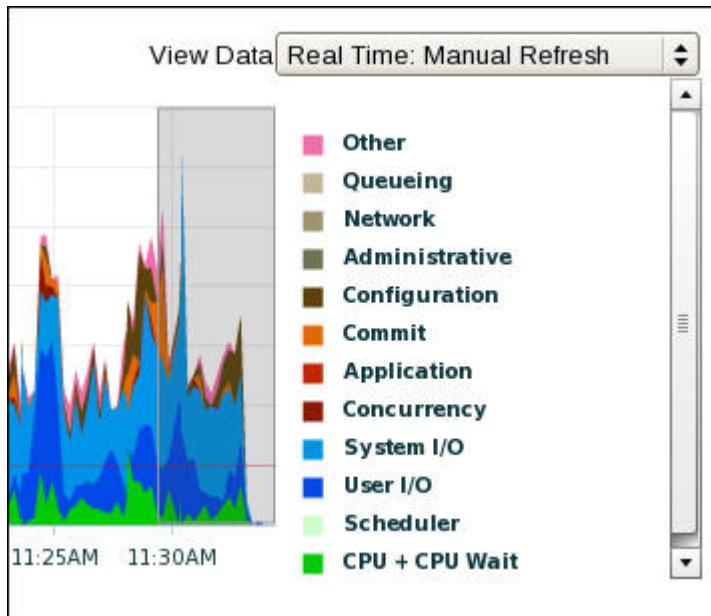
Start Time Jul 9, 2009 11:21:23 AM

Activity (%)	SID	QC SID	User	Program	Service	Plan Hash Value
100.00	86	SYS	sqlplus@edrsr12p1.us.oracle.com (TNS V1-V3)	SYS\$USERS	2494303166	

- b) Session Details 페이지에서 **Kill Session**을 누른 다음 **Yes**를 눌러 확인합니다.
- 주:** 몇몇 Automatic Refresh가 완료되기를 기다리느라 이 Session Details 페이지에 너무 오래 머무르는 경우 "WARNING, Session has expired" 경고가 나타나거나 세션이 종료되도록 표시되었다는 SQL 오류가 나타날 수 있습니다. 이 경고는 이미 종료된 세션에 대한 정보를 Refresh하려고 한다는 의미입니다. 이 경고는 무시해도 됩니다.

연습 13-1: 성능 관리 (계속)

- c) 페이지 상단의 탐색 기록에서 **Top Activity**를 누릅니다. 데이터베이스에서 세션 작업이 현저히 감소된 것을 확인할 수 있습니다.



단원 14의 연습

현재 상황: 테스트/개발 단계에서 운영 단계로 orcl 데이터베이스를 이전할 준비가 되었습니다.

failure 또는 데이터 손실 가능성을 줄이도록 데이터베이스를 구성합니다. 이를 위해 다음 작업을 수행하십시오.

- 콘트롤 파일의 중복성을 보장하고 콘트롤 파일을 Trace로 백업합니다.
- Fast Recovery Area 구성을 검토합니다.
- 각 그룹에 적어도 두 개의 리두 로그 멤버가 있는지 확인합니다.
- 데이터베이스를 ARCHIVELOG 모드로 설정합니다.
- 중복 아카이브 로그 대상을 구성합니다.

Oracle Internal & Oracle Academy
Use Only

연습 14-1: Recovery를 위한 디비오에이스 구성

이 연습에서는 Failure 또는 데이터 손실 가능성을 줄이도록 데이터베이스를 구성합니다.

주: 이 연습은 다음에 진행되는 모든 백업 및 Recovery 연습의 선수 과정이므로 반드시 마쳐야 합니다.

- 먼저 **lab_14_01_01.sh** 스크립트를 실행하여 이후 연습을 수행하는 동안 시나리오에서 사용할 데이터를 추가로 생성합니다. 이 스크립트는 INVENTORY 테이블스페이스에서 테이블을 생성하고 데이터베이스에서 일부 기본 작업을 시뮬레이트합니다.

```
$ cd ~/labs  
$ ./lab_14_01_01.sh
```

- 중복성을 보장하기 위해 최소한 두 개의 콘트롤 파일이 있는지 확인합니다.
 - orcl** 데이터베이스에 대해 **SYSDBA** 룰의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
 - Server > Control Files(Storage 섹션)를 누릅니다.

The screenshot shows the 'Control Files' section of the Oracle Enterprise Manager interface. It has tabs for General, Advanced, and Record Section, with General selected. A 'Backup To Trace' button is visible. Below is a table for 'Control File Mirror Images':

Valid	File Name	File Directory
VALID	current.260.692191347	+DATA/orcl/controlfile/
VALID	current.256.692191347	+FRA/orcl/controlfile/

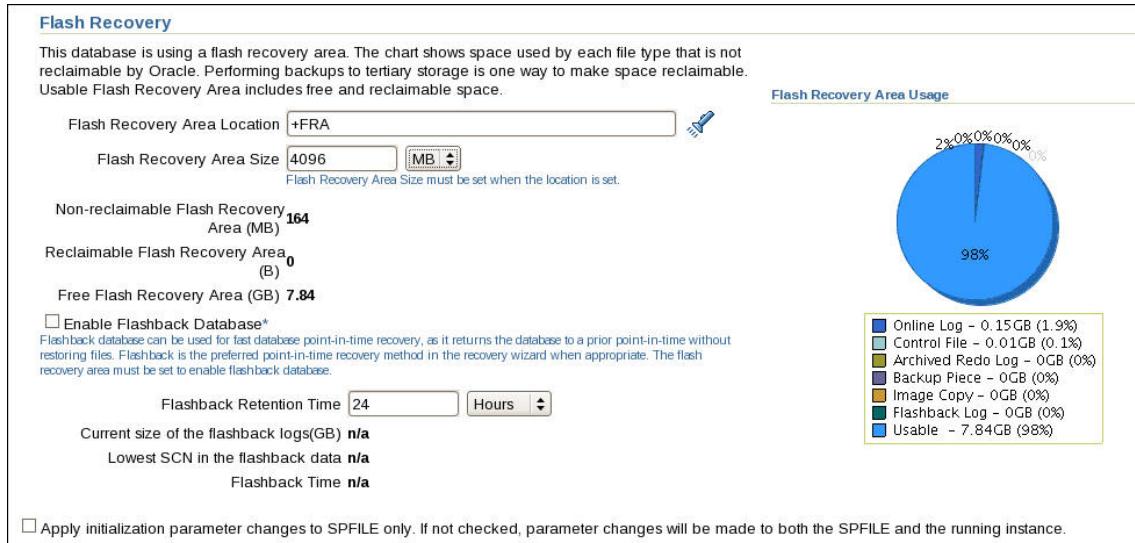
질문 1: Control Files: General 페이지에 몇 개의 콘트롤 파일이 있습니까?

해답: 2

연습 14-1: Recovery 를 위한 대이터베이스 구성 (계속)

3) Fast Recovery Area 구성을 검토하고 크기를 8GB로 변경합니다.

- Enterprise Manager의 Setup 섹션에서 Availability > Recovery Settings를 선택합니다.
- 페이지 아래쪽으로 이동합니다.



c) 질문: Fast Recovery Area가 활성화되어 있습니까?

해답: 예, 데이터베이스 생성 중 dbca를 사용하여 FRA가 구성되었기 때문입니다.

d) Fast Recovery Area의 위치를 기록해 둡니다.

예: +FRA

e) 질문: 필수 DBA 업무 중 이 섹션에서 수행할 수 있는 작업은 무엇입니까?

해답: Fast Recovery Area에 대한 위치, 크기 또는 Retention 시간을 변경하고 Flashback Database 기능을 활성화할 수 있습니다.

f) 질문: Fast Recovery Area 크기를 변경하면 데이터베이스를 재시작해야 합니까?

해답: 아니오, 이 변경 작업에는 재시작이 필요하지 않습니다.

g) "Flash Recovery Area Size" 필드에 8을 입력하고 size 필드 옆에 있는 선택 리스트에서 GB를 선택하여 Fast Recovery Area 크기를 8GB로 변경합니다.

h) 선택적으로 Show SQL을 눌러 명령문을 검토하고 Return을 누릅니다.

Show SQL	Return
ALTER SYSTEM SET db_recovery_file_dest_size = 8589934592 SCOPE=BOTH	

i) Apply를 누릅니다.

연습 14-1: Recovery 를 위한 데이터베이스 구성 (계속)

4) 각 리두 로그 그룹에 몇 개의 멤버가 있는지 확인합니다. 각 그룹에 적어도 두 개의 리두 로그 멤버가 있어야 합니다. 하나의 멤버 집합은 Fast Recovery Area에 저장해야 합니다.

- a) Server > Redo Log Groups를 누르고 "# of Members" 열에 몇 개의 멤버가 있는지 확인합니다.

해답: 각 그룹에 두 개의 멤버가 있습니다.

Redo Log Groups						
Object Type: Redo Log Group						
Search						
Enter an object name to filter the data that is displayed in your results set.						
Object Name	Go	Selection Mode	Single	Create	Actions	Clear logfile
Select Group	Status	# of Members	Archived	Size (KB)	Sequence	First Change#
<input checked="" type="radio"/> 1	Inactive	2	No	51200	127	3680499
<input type="radio"/> 2	Current	2	No	51200	128	3721156
<input type="radio"/> 3	Inactive	2	No	51200	126	3646837

- b) Redo Log Groups에서 하나를 선택하고 View를 눌러 해당 그룹의 각 멤버가 저장된 위치를 확인합니다. +DATA 디스크 그룹에 첫번째 멤버가 +FRA 디스크 그룹에 두번째 멤버가 보입니다.

View Redo Log Group: 1	
Actions: Clear logfile Go Edit Return	
Group # 1	
File size 51200 KB	
Status INACTIVE	
Redo Log Members	
File Name	File Directory
group_1.261.691672257	+DATA/orcl/onlinelog/
group_1.257.691672265	+FRA/orcl/onlinelog/

연습 14-1: Recovery 를 위한 데이터베이스 구성 (계속)

5) 각 로그 그룹에 대해 Archived 열은 No 값을 가집니다. 즉, 데이터베이스가 데이터베이스 Recovery에 사용할 리두 로그의 복사본을 보유하고 있지 않으므로 Failure 발생 시 마지막 백업 이후의 모든 데이터를 잃게 됩니다. 리두 로그가 아카이브되도록 데이터베이스를 ARCHIVELOG 모드로 합니다.

주: 변경 사항이 적용되도록 하려면 단계 5를 계속 진행해야 합니다.

- Enterprise Manager의 Setup 섹션에서 Availability > Recovery Settings를 선택합니다.
- Media Recovery 영역에서 **ARCHIVELOG Mode** 체크 박스를 선택합니다.
또한 Log Archive Filename Format에 %t, %s 및 %r이 포함되었는지 확인합니다.

Media Recovery

The database is currently in NOARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time are possible, but you must provide space for archived redo log files. If you change the database to ARCHIVELOG mode, you should perform a backup immediately. In NOARCHIVELOG mode, only cold backups are possible and data may be lost in the event of database corruption.

ARCHIVELOG Mode*

Log Archive Filename Format*

Number	Archived Redo Log Destination	Status	Type
1	USE_DB_RECOVERY_FILE_DEST	VALID	Local

Add Another Row

TIP It is recommended that archived redo log files be written to multiple locations spread across the different disks.
TIP You can specify up to 10 archived redo log destinations.

- 아카이브 로그 대상에 대한 현재 구성은 Fast Recovery Area(+FRA)를 가리키는 USE_DB_RECOVERY_FILE_DEST입니다.
주: 아카이브 로그 대상을 추가하는 경우 아직 디렉토리가 없으면 디렉토리를 생성해야 합니다.
- Apply를 누릅니다.
- 지금 데이터베이스를 재시작할지 묻는 메시지가 나타나면 Yes를 누릅니다.
- 호스트 인증서(Host Credentials)를 입력하여 데이터베이스(Username 및 암호로 oracle)를 재시작한 다음 OK를 누릅니다.
- 확인을 묻는 메시지가 나타나면 Yes를 다시 누릅니다.
- 종료 및 시작 작업 중 오류 메시지가 나타나면 OK를 눌러 오류를 확인한 다음 Refresh를 다시 누릅니다. (유저 작업 속도가 데이터베이스보다 빨랐을 수 있습니다.)

연습 14-1: Recovery 를 위한 대이터베이스 구성 (계속)

- 6) 선택적으로 데이터베이스가 재시작되면 SQL*Plus를 사용하여 데이터베이스가 ARCHIVELOG 모드인지 확인합니다. 터미널 window에서 SQL*Plus에 **SYSDBA**로 로그인하고 **archive log list** 명령을 실행합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on
Sat Jul 11 10:16:40 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> archive log list
Database log mode          Archive Mode
Automatic archival        Enabled
Archive destination        USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 126
Next log sequence to archive 128
Current log sequence       128
SQL> exit
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
$
```

이제 데이터베이스가 ARCHIVELOG 모드에서 각 온라인 리두 로그 파일의 복사본을 계속 아카이브하므로 추가 리두 데이터에 대해 파일을 재사용할 수 있습니다.

주: 이 작업은 디스크 공간을 사용하므로 오래된 아카이브 로그는 다른 저장 영역으로 정기적으로 백업해야 합니다.

단원 15의 연습

현재 상황: 데이터베이스를 운영할 준비가 되었으므로 이제 백업을 시작합니다.
추적할 콘트롤 파일을 백업하고 디스크에 즉시 백업을 수행하고, 무한으로 반복되는
야간 백업 작업의 일정을 잡습니다.

Oracle Internal & Oracle Academy
Use Only

연습 15-1: 데이터베이스 백업 수행

이 연습에서는 추적할 콘트롤 파일을 백업하고, 디스크에 즉시 백업을 수행하고, 야간 백업 작업의 일정을 잡습니다.

- 1) 추적할 콘트롤 파일의 백업을 수행합니다.

- a) **orcl** 데이터베이스에 대해 **SYSDBA** 를의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
- b) **Server > Control Files(Storage 섹션)**를 누릅니다.

The screenshot shows the 'Control Files' section of the Oracle Enterprise Manager interface. It displays two control files listed in a table:

Valid	File Name	File Directory
VALID	current.260.692191347	+DATA/orcl/controlfile/
VALID	current.256.692191347	+FRA/orcl/controlfile/

A 'Backup To Trace' button is visible at the top right of the table area.

- c) **Backup to Trace**를 누릅니다.
- d) 성공 메시지가 나타나면 Trace 디렉토리 위치를 확인한 다음 **OK**를 누릅니다.

The screenshot shows an 'Update Message' window with the following text:
Control file successfully backed up to trace at /u01/app/oracle/diag/rdbms/orcl/orcl/trace
An 'OK' button is at the bottom right.

- e) 선택적으로 oracle 유저로 로그인한 터미널 window에서 다음 명령을 실행하여 Alert Log 끝에서 Trace File 이름을 확인합니다.

```
cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
tail alert_orcl.log
```

다음 출력은 마지막 몇 행만 표시한 것입니다.

```
$ cd /u01/app/oracle/diag/rdbms/orcl/orcl/trace
$ tail alert_orcl.log
Sat Jul 11 09:10:03 2009
SMCO started with pid=23, OS id=9837
Sat Jul 11 09:46:31 2009
ALTER DATABASE BACKUP CONTROLFILE TO TRACE
Backup controlfile written to trace file
/u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl_ora_12190.trc
Completed: ALTER DATABASE BACKUP CONTROLFILE TO TRACE
Sat Jul 11 09:46:56 2009
ALTER DATABASE BACKUP CONTROLFILE TO TRACE
Backup controlfile written to trace file
/u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl_ora_12190.trc
Completed: ALTER DATABASE BACKUP CONTROLFILE TO TRACE
$
```

연습 15-1: 데이터베이스 백업 수행 (계속)

- f) 선택적으로 콘트롤 파일 내에서 다른 섹션의 크기 및 사용을 보려면 Record Section 탭 페이지를 누릅니다.

The screenshot shows the Oracle Enterprise Manager 11g interface. The title bar says 'ORACLE Enterprise Manager 11g Database Control'. The top menu has items like Setup, Preferences, Help (which is highlighted with a red box), Logout, and Database. Below the menu, it says 'Logged in As SYS'. The main content area is titled 'Control Files' and has tabs for General, Advanced, and Record Section (which is also highlighted with a red box). Under 'Control File Record Information', there is a table with columns Type, Record Size, Records Total, and Records Used. The table lists various database components with their respective record sizes and counts.

Type	Record Size	Records Total	Records Used
CKPT PROGRESS	8180	11	0
DATABASE	316	1	1
DATAFILE	520	100	7
FILENAME	524	2298	14
LOG HISTORY	56	292	5
REDO LOG	72	16	3
REDO THREAD	256	8	1
TABLESPACE	68	100	7

유저마다 숫자는 다르게 나타날 수 있습니다. 추가 정보를 보려면 페이지 오른쪽 상단 모서리에서 Help를 누릅니다.

- 2) 백업 셋과 Image Copy의 차이는 무엇입니까?

해답: 백업 셋에는 오라클 고유 형식으로 채워진 아카이브 로그 파일과 데이터가 있습니다. 사용하기 전에 파일을 추출해야 합니다. Image Copy는 운영 체제 파일 복사본과 같으므로 이러한 복사본을 사용하여 작업을 즉시 복원할 수 있습니다.

- 3) 디스크 백업은 어느 대상에서 수행됩니까?

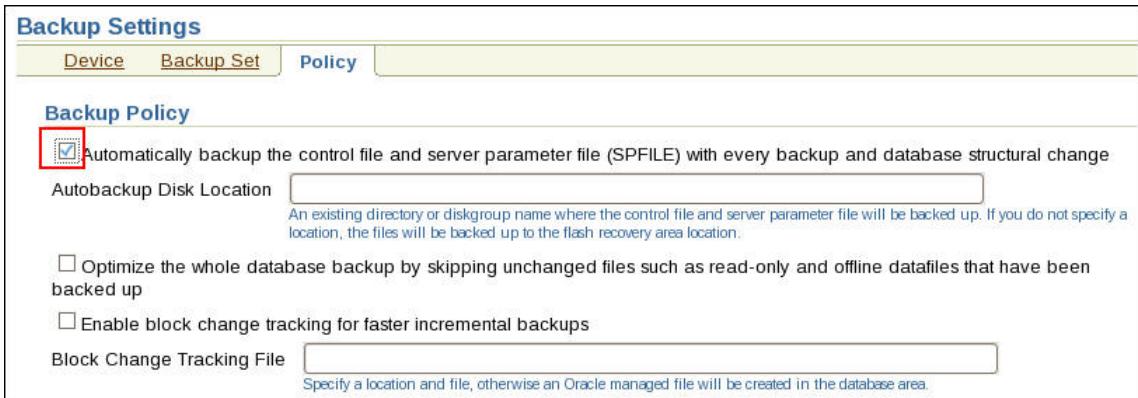
- a) Availability 페이지로 이동하여 Backup Settings를 누릅니다.

The screenshot shows the Oracle Enterprise Manager 11g Backup Settings page. The title bar says 'Backup Settings'. The top menu has tabs for Device, Backup Set, and Policy. The main content area is titled 'Disk Settings' and includes fields for Parallelism (set to 1), Disk Backup Location, and Disk Backup Type. The Disk Backup Type section has three options: Backup Set (selected), Compressed Backup Set, and Image Copy. A note below the Disk Backup Location field states: 'The flash recovery area is the current disk backup location. If you would like to override the disk backup location, specify an existing directory or diskgroup.' There is also a 'Test Disk Backup' button.

- b) Disk Backup Location 아래의 메시지는 Fast Recovery Area가 현재 디스크 백업 위치임을 알려줍니다.

연습 15-1: 디비오에이스 백업 수행 (계속)

- 4) SPFILE 및 콘트롤 파일을 자동으로 백업하도록 백업 정책을 설정합니다.
- Backup Settings 제목에서 **Policy** 탭을 누릅니다.
 - "Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change"를 선택합니다.



- 아래쪽으로 이동하여 서버의 Host Credentials Username 및 Password에 각각 **oracle**을 입력하고 "Save as Preferred Credential"을 선택합니다.
 - OK를 누릅니다.
- 5) Host Credentials에 oracle을 사용하여 백업 세트으로 디스크 백업을 테스트합니다.
- Backup Settings 페이지에서 **Device** 탭을 누릅니다.
 - Disk Backup Type으로 **Backup Set**을 선택합니다.
 - 아래쪽으로 이동하여 Host Credentials가 oracle로 설정되어 있는지 확인합니다.
 - 페이지 맨 위로 이동하여 **Test Disk Backup**을 누릅니다.
 - 처리 메시지가 나타납니다. 테스트가 종료되고 "Disk Backup Test Successful!" 메시지가 표시되면 페이지 아래쪽으로 이동하고 OK를 누릅니다.
- 6) 데이터베이스가 유저 작업용으로 열려 있는 동안 아카이브 로그와 함께 전체 데이터베이스를 백업합니다. 이 백업은 Incremental 백업 전략의 기반으로 사용되어야 합니다.
- 질문: 데이터베이스를 종료하지 않고 유효한 데이터베이스 백업을 생성하기 위해 반드시 충족해야 하는 필요 조건은 무엇입니까?
해답: 데이터베이스가 ARCHIVELOG 모드여야 합니다. 데이터베이스가 열려 있지만 ARCHIVELOG 모드가 아닌 상태에서 수행된 백업은 Recovery에 사용할 수 없습니다.

연습 15-1: 데이터베이스 백업 수행 (계속)

- b) Availability > Schedule Backup(Manage 섹션)을 선택합니다.
- Oracle-Suggested Backup 전략이 자신의 요구와 정확히 일치하면 이 옵션을 선택합니다. 그러나 여기에서는 연습 목적으로 커스터마이즈된 백업의 일정을 잡습니다.
- c) 백업 할 객체로 Whole Database를 선택합니다.
- d) 서버의 Host Credentials Username 및 Password에 각각 **oracle**을 확인하거나 입력합니다.
- e) Schedule Customized Backup을 누릅니다.
- f) Schedule Customized Backup: Options 페이지에서 Backup Type에 **Full Backup**을 선택하고 "Use as the base of an incremental backup strategy" 체크 박스를 선택합니다.
- g) Backup Mode로 Online Backup을 선택합니다.
- h) Advanced 섹션에서 "Also back up all archived logs on disk" 및 "Delete all archived logs from disk after they are successfully backed up"을 선택한 다음 Next를 눌러 계속합니다.
- i) Schedule Customized Backup: Settings 페이지에서 백업 위치에 Disk를 선택합니다. (Disk Backup Location을 그대로 두고 일회용 백업을 위해 현재 설정을 무효화할 수 있습니다. 그러나 이번에는 해당 버튼을 누르지 마십시오.)

Schedule Customized Backup: Settings

Database orcl.oracle.com
Backup Strategy Customized Backup
Object Type Whole Database

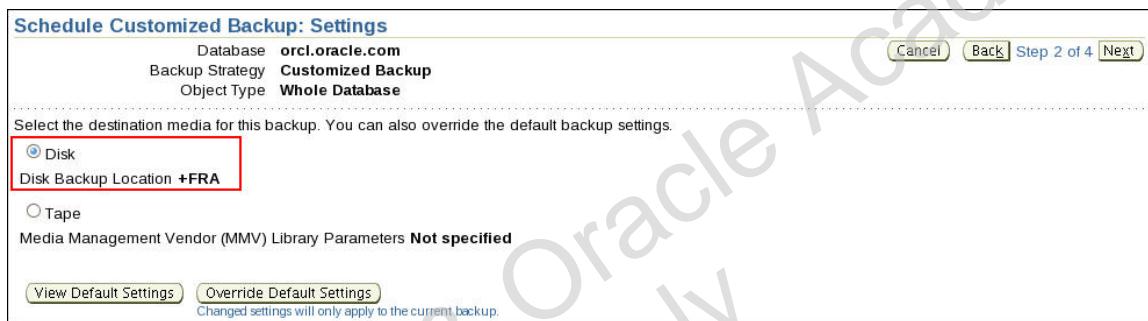
Select the destination media for this backup. You can also override the default backup settings.

Disk
 Tape

Disk Backup Location +FRA
Media Management Vendor (MMV) Library Parameters Not specified

[View Default Settings](#) [Override Default Settings](#)
Changed settings will only apply to the current backup.

Cancel Back Step 2 of 4 Next



- j) Next를 누릅니다.
- k) Schedule Customized Backup: Schedule 페이지에서 기본값을 모두 적용한 다음 Next를 눌러 진행합니다.
주: Schedule Type은 One Time(Immediately)이어야 합니다.

연습 15-1: 데이터베이스 백업 수행 (계속)

- l) Schedule Customized Backup: Review 페이지에서 RMAN 스크립트를 검토한 다음 Submit Job을 누릅니다.

Schedule Customized Backup: Review

Database: orcl.oracle.com
Backup Strategy: Customized Backup
Object Type: Whole Database

Settings

Destination: Disk
Backup Type: Use as the base of an incremental backup strategy
Backup Mode: Online Backup
Flash Recovery Area: +FRA

RMAN Script

The RMAN script below is generated based on previous input.

```
backup incremental level 0 cumulative device type disk tag '%TAG' database;
backup device type disk tag '%TAG' archivelog all not backed up delete all input;
```

- m) View Job을 눌러 백업 작업의 상태를 모니터합니다. 이 백업 시간은 하드웨어 및 시스템 리소스에 따라 다릅니다.
- n) 작업이 완료될 때까지 브라우저의 Refresh 또는 Requery 버튼을 누릅니다.

Name	Targets	Status	Started	Ended	Elapsed Time (seconds)
Execution: orcl.oracle.com	orcl.oracle.com	Succeeded	Jul 11, 2009 10:28:05 AM (UTC+07:00)	Jul 11, 2009 10:34:32 AM (UTC+07:00)	387
Step: Prebackup	orcl.oracle.com	Succeeded	Jul 11, 2009 10:28:25 AM (UTC+07:00)	Jul 11, 2009 10:28:26 AM (UTC+07:00)	1
Step: Backup	orcl.oracle.com	Succeeded	Jul 11, 2009 10:28:35 AM (UTC+07:00)	Jul 11, 2009 10:34:21 AM (UTC+07:00)	346
Step: Post Backup	orcl.oracle.com	Succeeded	Jul 11, 2009 10:34:30 AM (UTC+07:00)	Jul 11, 2009 10:34:32 AM (UTC+07:00)	2

- 7) 아카이브 로그를 포함하여 전체 데이터베이스에 대해 야간에 실행되는 디스크 기반 Incremental 온라인 백업의 일정을 잡습니다. 백업이 완료되면 디스크에서 아카이브 로그를 삭제하도록 설정합니다. 11:00 PM에 실행되도록 일정을 잡습니다. 이 스케줄은 무기한 유효해야 합니다.

- Enterprise Manager에서 Availability > Schedule Backup(Manage 섹션)을 선택합니다.
- 백업 할 객체로 Whole Database를 선택합니다.
- 서버의 Host Credentials Username 및 Password에 각각 **oracle**을 확인하거나 입력한 다음 Schedule Customized Backup을 누릅니다.

연습 15-1: 데이터베이스 백업 수행 (계속)

- d) Schedule Customized Backup: Options 페이지에서 Backup Type으로 **Incremental Backup**을 선택합니다.
- e) Backup Mode로 **Online Backup**을 선택합니다.
- f) Advanced 영역에서 "Also backup all archived logs on disk" 및 "Delete all archived logs from disk after they are successfully backed up"을 선택한 다음 **Next**를 눌러 계속합니다.
- g) Schedule Customized Backup: Settings 페이지에서 백업 위치로 **Disk**를 선택한 다음 **Next**를 눌러 계속 진행합니다.
- h) Schedule Customized Backup: Schedule 페이지에서 Job Name을 **Nightly_Backup**으로 변경하고 Job Description에 기본값을 적용합니다.
- i) Schedule 영역에서 **Repeating**을 선택합니다. 컨텍스트에 따른 추가 세부 정보가 표시됩니다.
- j) Frequency Type drop-down list에서 **By Days**를 선택하고, Repeat Every 필드에 **1**을 입력하고, Repeat Until 값으로 Indefinite가 선택되어 있는지 확인하고, Start Time으로 **11:00 PM**을 입력합니다.

Schedule Customized Backup: Schedule

Database: orcl.oracle.com
Backup Strategy: Customized Backup
Object Type: Whole Database

Job

* Job Name: Nightly_Backup
Job Description: Whole Database Backup

Schedule

Type: One Time (Immediately) One Time (Later) Repeating

Frequency Type: By Days

Repeat Every: 1 Days

Time Zone: (UTC-06:00) US Central Time (CST)

Start Date: Jul 11, 2009

Start Time: 11:00 AM

Repeat Until: Indefinite
 Specified Date
Date: (example: Jul 11, 2009)
Time: : AM

Buttons: Cancel, Back, Step 3 of 4, Next

연습 15-1: 데비아리스 백업 수행 (계속)

- k) Next를 눌러 계속합니다.
- l) Schedule Customized Backup: Review 페이지에서 설정과 RMAN 스크립트를 검토합니다.

Schedule Customized Backup: Review

Database	orcl.oracle.com	(Cancel)	Edit RMAN Script	Back	Step 4 of 4	Submit Job
Backup Strategy	Customized Backup					
Object Type	Whole Database					
Settings						
Destination	Disk					
Backup Type	Incremental Backup					
Backup Mode	Online Backup					
Flash Recovery Area	+FRA					
RMAN Script						
The RMAN script below is generated based on previous input.						
backup incremental level 1 cumulative device type disk tag '%TAG' database; backup device type disk tag '%TAG' archivelog all not backed up delete all input;						

- m) Submit Job을 누른 다음 OK를 누릅니다.
- n) Availability 페이지의 Related Links 섹션에서 Jobs를 누르고 Job Activity list에서 일정이 잡힌 작업을 확인합니다.

View	Runs	OS Command													
		View Results		Edit		Create Like		Copy To Library		Suspend	Resume	Stop	Delete	Create Job	Go
Select	Name	Status	(Executions)	Scheduled	Targets		Target Type	Owner	Job Type						
<input checked="" type="radio"/>	NIGHTLY_BACKUP	1 Scheduled		Jul 11, 2009 11:00:00 PM CDT	orcl.oracle.com	Database Instance	SYS	Database Backup							
<input type="radio"/>	BACKUP_ORCL.ORACLE.COM_000003	1 Succeeded		Jul 11, 2009 10:28:05 AM (UTC+07:00)	orcl.oracle.com	Database Instance	SYS	Database Backup							
<input type="radio"/>	REORGANIZE_ORCL.ORACLE.COM_1	1 Succeeded		Jul 9, 2009 10:52:46 AM (UTC+07:00)	orcl.oracle.com	Database Instance	SYS	Reorganize							

단원 16의 연습

현재 상황: 오라클 데이터베이스 failure 중 많은 부분의 원인을 디스크나 컨트롤러 failure와 같은 일종의 Media failure에서 찾을 수 있습니다. 이 연습에서는 데이터베이스 Recovery를 요하는 여러 가지 문제를 경험하게 됩니다.

- 손실된 데이터 파일을 Recovery합니다.
- 손실된 SYSTEM 테이블스페이스의 파일을 Recovery합니다.
- 손실된 콘트롤 파일을 Recovery합니다.

SQL 스크립트 파일은 /home/oracle/labs 디렉토리에 제공되어 있습니다.

필요한 경우 Linux 및 SQL 구문에 대한 부록을 참조하십시오. SQL 스크립트로 Failure를 설정한 후에는 다른 연습을 진행하기 전에 반드시 Recovery를 완료해야 합니다.

주: 시스템에 따라 OS 파일 이름은 여기에 표시된 것과 다를 수 있습니다. 유저마다 출력은 다르게 나타날 수 있습니다. (공간을 절약하기 위해 빈 줄은 제거했습니다.)

Oracle Internal & Oracle Academy
Use Only

연습 16-1: 연습 환경 준비

Recovery 시나리오 중 하나를 시작하기 전에 남은 Recovery 연습을 위한 환경을 준비하는 스크립트를 실행해야 합니다.

- 1) 개별 문제를 설정하기 전에 **labs** 디렉토리로 이동하여 SQL*Plus에서 **SYS** 유저로 **lab_16_01_01.sql** 스크립트를 실행해야 합니다. 이 스크립트는 이 연습의 나머지 부분에서 호출할 일부 프로시저를 준비합니다.

```
$ cd /home/oracle/labs
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on
Sun Jul 12 23:29:58 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> @lab_16_01_01.sql
Connected.

Java created.

Procedure created.

PL/SQL procedure successfully completed.

Synonym created.

Grant succeeded.

SQL>
```

연습 16-2: *DFO/EI 파일 손실 Recovery*

이 연습에서는 응용 프로그램 데이터 파일이 손실되는 문제가 발생합니다. 그런 다음 이 손실을 Recovery하기 위한 단계를 수행합니다.

- 1) SQL*Plus 세션에서 SYS 유저로 labs 디렉토리에서 **lab_16_02_01.sql** 스크립트를 실행합니다. 이 스크립트는 응용 프로그램 데이터 파일 중 하나를 삭제합니다.

주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 **orcl** 데이터베이스에 맞게 설정합니다.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.1.0.6.0 - Production on
Sun Aug 31 10:45:19 2008

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
Production
With the Partitioning, OLAP, Data Mining and Real
Application Testing options

SQL> @lab_16_02_01.sql

PL/SQL procedure successfully completed.

"Data file deleted. Database should be open."
SQL>
```

- 2) Help Desk에서 HR 응용 프로그램 스키마의 COUNTRIES 테이블에 액세스할 수 없다는 신고 전화가 접수되었습니다. 테이블의 행 개수를 세고 문제가 있는지 확인합니다.
주: 데이터 파일 오류가 표시되지 않으면 다른 SQL*Plus 세션에서 INVENTORY.PRODUCT_MASTER 테이블에 대해 갱신을 수행해봅니다. 오라클 데이터베이스에서 파일이 손실되었음을 인식하기까지는 약간의 시간이 걸릴 수 있습니다. 기다리는 것을 원하지 않을 경우 이 단계를 건너뛰고 다음 단계로 이동할 수 있습니다.

```
SQL> select count(*) from INVENTORY.PRODUCT_MASTER;
select count(*) from INVENTORY.PRODUCT_MASTER
*
ERROR at line 1:
ORA-00376: file 7 cannot be read at this time
ORA-01110: data file 7:
'/u01/app/oracle/oradata/orcl/inventory02.dbf'
SQL>
```

연습 16-2: *DFO/E* 파일 손실 Recovery(계속)

- 3) 문제를 해결하고 필요한 경우 recovery합니다. 오류 메시지는 inventory02.dbf 데이터 파일이 손상되었거나 누락되었다고 설명합니다.
- Enterprise Manager의 Home 페이지에서 Alerts 섹션을 조사하고 Data Failure alert를 확인합니다.
 - Availability > Perform Recovery**를 누릅니다.
 - Advise and Recover**를 누릅니다.
 - View and Manage Failures 페이지에서 Failure Description 아래의 더하기(+) 아이콘을 누릅니다. 다음과 같은 Failure가 나타나야 합니다.

Select Failure Description	Impact	Priority	Status	Time Detected
<input type="checkbox"/> Data Failures				
<input checked="" type="checkbox"/> One or more non-system datafiles are missing	See impact for individual child failures	HIGH	OPEN	2009-07-14 13:35:43.0
<input checked="" type="checkbox"/> Datafile 7: '/u01/app/oracle/oradata/orcl/inventory02.dbf' is missing	Some objects in tablespace INVENTORY might be unavailable	HIGH	OPEN	2009-07-14 13:35:43.0

주: 비시스템 데이터 파일 Failure가 나타나지 않으면 표시될 때까지 페이지 refresh를 계속 수행합니다.

- Failure가 선택된 상태에서 **Advise**를 누릅니다.
- 파일 이름을 바꾸거나 이동한 것이 아니라 파일이 삭제된 것으로 "Continue with Advise"를 눌러 계속 진행합니다.

The following user actions may provide a faster recovery path for certain simple failures. Click "Re-assess Failures" if user actions are performed. Otherwise, click "Continue with Advise" to use the recovery advice generated for the failures selected.

Manual Action Details
If file '/u01/app/oracle/oradata/orcl/inventory02.dbf' was unintentionally renamed or moved, restore it.

- Recovery Advice 페이지에서 RMAN 스크립트를 확인합니다. **Continue**를 누릅니다.

The repair includes complete media recovery with no data loss

RMAN Script

```
# restore and recover datafile
sql 'alter database datafile 7 offline';
restore datafile 7;
recover datafile 7;
sql 'alter database datafile 7 online';
```

연습 16-2: DFO/EI 파일 손실 Recovery(계속)

- h) Review 페이지에서 Failure와 제안된 해결책을 확인합니다. "Submit Recovery Job"을 누릅니다.
- i) Processing window가 나타나고 이어서 Job Activity 페이지가 나타납니다. 작업이 성공적으로 생성되었다는 메시지가 나타나야 합니다. (링크 이름은 다를 수 있습니다.)



- j) 작업 이름 링크를 누릅니다.
- k) Job Run 페이지의 Summary 섹션에서 Status를 확인합니다. Running 상태인 경우 작업이 완료될 때까지 브라우저의 Refresh 또는 Requery 버튼을 사용합니다.

Summary	
Status	Succeeded
Scheduled	Jul 14, 2009 1:44:35 PM (UTC+07:00)
Started	Jul 14, 2009 1:44:35 PM (UTC+07:00)
Ended	Jul 14, 2009 1:45:18 PM (UTC+07:00)
Elapsed Time	42 seconds
Notification	No

- l) SQL*Plus 세션에서 INVENTORY.PRODUCT_MASTER 테이블에 지금 액세스할 수 있는지 확인합니다.
주: COUNT 값은 여기에 나오는 것과 다를 수 있습니다.

```
SQL> select count(*) from INVENTORY.PRODUCT_MASTER;

COUNT(*)
-----
217368
```

- m) Recovery가 완료되었으므로 Enterprise Manager에서 데이터베이스 Home 페이지로 이동하고 alert를 지웁니다.
 - i) Message 열에서 Data Failure alert에 대한 링크를 누릅니다.
 - ii) Clear Open Alert를 누릅니다.

연습 16-3: SYSTEM 테이블스페이스에서 파일 손실 Recovery

이 연습에서는 시스템에서 SYSTEM 테이블스페이스의 파일 손실이 발생합니다.
그런 다음 이 손실을 Recovery하기 위한 단계를 수행합니다.

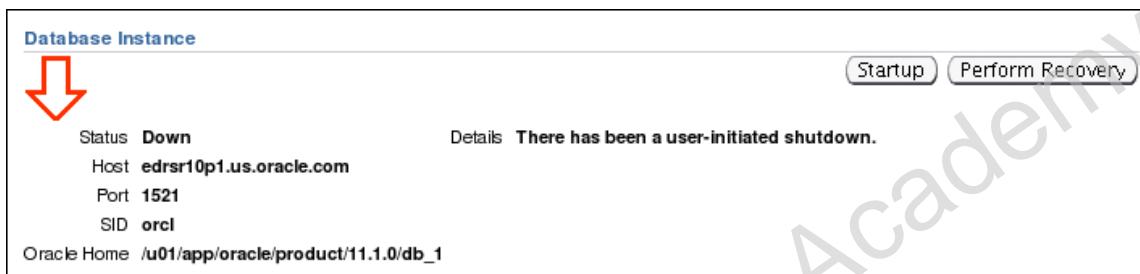
- 1) 시스템 데이터 파일 또는 언두 데이터 파일스페이스에 속하는 데이터 파일의 손상을 Recovery하는 것이 응용 프로그램 데이터 파일의 Recovery와 다른 이유는 무엇입니까?

해답: 시스템이나 언두 데이터 파일의 Recovery는 데이터베이스를 닫은 상태에서 수행되어야 하는 반면 응용 프로그램 데이터 파일의 Recovery는 데이터베이스를 열어 두고 다른 유저가 사용할 수 있는 상태에서 수행되어야 합니다.

- 2) oracle OS 유저로 labs 디렉토리에서 **lab_16_03_02.sh** 스크립트를 실행합니다. 이 스크립트는 시스템 데이터 파일을 삭제합니다.

```
$ ./lab_16_03_02.sh
```

- 3) Enterprise Manager에서 Database home 페이지를 검토합니다. 연결이 거부되었다는 메시지가 나타나면 브라우저에 EM 홈 페이지 URL을 다시 입력해 봅니다.
Database home 페이지를 표시하려면 여러 번 시도해야 할 수 있습니다.



- 4) 데이터베이스가 종료되었습니다. 데이터베이스를 시작해 봅니다.
 - a) **Startup**을 눌러 열기를 시도합니다.
 - b) "Startup/Shutdown: Specify Host and Target Database Credentials" 페이지에서 Host Credentials로 **oracle** 및 **oracle**을 입력합니다. **OK**를 누릅니다.
 - c) Select Startup Type 페이지에서 기본 선택 항목을 그대로 두고 **OK**를 누릅니다.
 - d) Startup/Shutdown:Confirmation 페이지에서 **Yes**를 누릅니다.

연습 16-3: SYSTEM 테이블스페이스에서 파일 손실 Recovery (계속)

- e) 데이터 파일 1을 식별할 수 없다는 Srvctl Error가 표시됩니다. Database 탭을 눌러 데이터베이스 Home 페이지로 돌아갑니다.

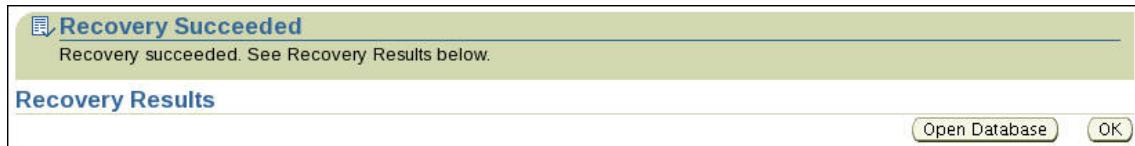
The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. At the top, there's a navigation bar with 'ORACLE Enterprise Manager 11g' and 'Database Control'. A red box highlights the 'Database' tab in the top right corner. Below the navigation bar, a modal dialog box is open with the title 'Srvctl Error'. The error message inside the dialog reads: 'An error occurred when executing the operation. Ensure that CRS processes and SRVCTL are functioning properly. Refresh the page to see the current status. Refer to the following error: startup open PRCR-1079 : Failed to start resource ora.orcl.db ORA-01157: cannot identify/lock data file 1 - see DBWR trace file ORA-01110: data file 1: '+DATA/orcl/datafile /system.256.692191147' CRS-2674: Start of 'ora.orcl.db' on 'edrsr12p1' failed'. Below the dialog, the main interface shows a 'Startup/Shutdown: Select Operation' section with tabs for 'Startup' and 'Shutdown'. It says 'Start up or shut down the database managed by Oracle Restart.' and 'Page Refreshed 7/14/09 1:58 PM'. There's a 'Refresh' button. Below this, there's a table with columns 'Select', 'Instance Name', 'Status', and 'Host Name'. A single row is selected for 'orcl', which has a red downward arrow icon next to it under 'Status' and 'n/a' under 'Host Name'. A red box highlights the 'Select' column header.

- 5) Perform Recovery를 누릅니다.

- Host Credentials로 **oracle** 및 **oracle**을 입력하고 Continue를 누릅니다.
- 데이터베이스가 현재 중지되어 있으므로 Recovery를 수행하려면 마운트된 상태로 시작되어야 합니다. Start up the Database 링크를 누릅니다.
 - 호스트 인증서(Host Credentials)를 다시 입력하라는 메시지가 표시되면 **oracle** 및 **oracle**을 입력하고 OK를 누릅니다.
 - Select Startup Type 페이지에서 기본 선택 항목을 그대로 두고 OK를 누릅니다.
 - Startup/Shutdown: Confirmation 페이지에서 Advanced Options를 누릅니다.
 - Startup 모드로 Mount the database를 선택하고 OK를 누릅니다.
 - Startup/Shutdown: Confirmation 페이지에서 Yes를 누릅니다.
 - 시작 작업이 성공합니다. Database 탭을 누릅니다.
- 이제 데이터베이스는 마운트된 상태입니다. Perform Recovery를 다시 누릅니다.
- Host Credentials 섹션에서 username 및 암호에 대해 **oracle**을 입력합니다. 그런 다음 Continue를 누릅니다.
- Database Login 페이지에서 username에 **dba1**, 암호에 **oracle_4U**, Connect As에 **SYSDBA**를 입력합니다. 그런 다음 Login을 누릅니다.
- Perform Recovery 페이지에서 Advise and Recover를 누릅니다.
- View and Manage Failures 페이지에서 Failure에 대한 설명을 검토합니다. 시스템 데이터 파일 Failure가 선택되었는지 확인한 다음 Advise를 누릅니다.
- Manual Actions 페이지에서 Continue with Advise를 누릅니다.

연습 16-3: SYSTEM 테이블스페이스에서 파일 손실 Recovery (계속)

- i) Recovery Advice 페이지에서 RMAN 스크립트를 검토한 다음 **Continue**를 누릅니다.
- j) Review 페이지에서 **Execute Recovery**를 누릅니다.
- k) 처리 페이지가 나타나고 이어서 Perform Recovery: Result 페이지가 나타납니다. 이 작업 기간은 시스템 리소스에 따라 다릅니다. Recovery 작업을 성공해야 합니다.



- l) Perform Recovery: Result 페이지에서 **Open Database**를 누릅니다.
- m) 성공 메시지가 나타난 후 **OK**를 누릅니다.
- n) **DBA1** 유저 (**SYSDBA**)로 EM에 로그인하고 Database home 페이지를 검토하여 데이터베이스가 열려 있고 정상적으로 작동하는지 확인합니다.
- o) Recovery가 완료되었으므로 alert를 지웁니다.
 - i) Message 열에서 Data Failure alert에 대한 링크를 누릅니다.
 - ii) **Clear Open Alert**를 누릅니다.

연습 16-4: 콘트롤 파일 손실 Recovery

이 연습에서는 시스템에서 콘트롤 파일 손실이 발생합니다. RMAN 명령행 환경을 소개하기 위해 이 연습에서는 수동 Recovery 프로세스를 안내합니다.

- 터미널 window에서 oracle 유저로 **lab_16_04_01.sh** 스크립트를 실행합니다. 이 스크립트는 콘트롤 파일 중 하나를 삭제합니다.

```
$ ./lab_16_04_01.sh
```

- Help Desk에 데이터베이스가 다운된 것 같다는 문의 전화가 걸려오기 시작합니다. 문제를 해결하고 필요한 경우 recovery합니다. SRVCTL을 사용하여 데이터베이스를 시작합니다.
 - 터미널 window에서 **oraenv**를 사용하여 유저 환경이 **orcl** 데이터베이스 환경에 맞게 구성되었는지 확인합니다.
 - 다음 SRVCTL 명령을 실행하여 데이터베이스를 시작합니다.

```
srvctl start database -d orcl -o open
```

```
$ srvctl start database -d orcl -o open
PRCR-1079 : Failed to start resource ora.orcl.db
ORA-00205: error in identifying control file, check alert
log for more info
CRS-2674: Start of 'ora.orcl.db' on 'edrsr12p1' failed
```

- 콘트롤 파일을 식별하는 오류에 주의합니다.
- alert_orcl.log**
파일(**/u01/app/oracle/diag/rdbms/orcl/orcl/trace** 디렉토리에 있음)을 조사하여 이 Failure를 자세히 조사합니다. 터미널 window에서 다음을 입력하여 **alert_orcl.log** 파일의 마지막 20개 행을 조사합니다.

```
$ tail -20 alert_orcl.log
ALTER DATABASE MOUNT
NOTE: Loaded library:
/opt/oracle/extapi/32/asm/orcl/1/libasm.so
NOTE: Loaded library: System
SUCCESS: diskgroup DATA was mounted
SUCCESS: diskgroup FRA was mounted
ORA-00210: cannot open the specified control file
ORA-00202: control file:
'+FRA/orcl/controlfile/current.256.695531381'
ORA-17503: ksfdopn:2 Failed to open file
+FRA/orcl/controlfile/current.256.695531381
ORA-15012: ASM file
'+FRA/orcl/controlfile/current.256.695531381' does not exist
NOTE: dependency between database orcl and diskgroup
resource ora.DATA.dg is established
Sat Aug 22 03:18:05 2009
Checker run found 1 new persistent data failures
ORA-205 signalled during: ALTER DATABASE MOUNT...
Sat Aug 22 03:18:08 2009
```

연습 16-4: 콘트롤 파일 손실 Recovery (계속)

```
Shutting down instance (abort)
License high water mark = 1
USER (ospid: 17765): terminating the instance
Instance terminated by USER, pid = 17765
Sat Aug 22 03:18:08 2009
Instance shutdown complete
$
```

- 4) 누락된 콘트롤 파일은 **+FRA** 디스크 그룹에 있는 파일입니다. 또한 **+DATA** 디스크 그룹에도 콘트롤 파일이 하나 있음을 알고 있습니다. **+DATA** 디스크 그룹에 있는 콘트롤 파일로부터 복원하여 Recovery를 수행할 수 있지만 그러면 파일 이름을 알아야 합니다. **asmcmd**를 사용하여 **+DATA** 디스크 그룹에 있는 콘트롤 파일의 이름을 확인합니다.
- a) 새 터미널 window를 열고 **oraenv**를 사용하여 **+ASM** Instance에 맞게 환경을 설정합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
```

- b) **asmcmd**를 시작하고 **ls** 명령을 사용하여 **+DATA** 디스크 그룹에서 콘트롤 파일(이 파일은 **+data/orcl/controlfile** 디렉토리에 있음)의 이름을 확인합니다.

```
$ asmcmd
ASMCMD> ls +data/orcl/controlfile
Current.260.695209463
ASMCMD>
```

- c) 다음 단계에서 참조할 수 있도록 이 이름과 전체 경로를 적어둡니다.
- 5) 또 다른 터미널 window에서 **RMAN**에 연결하고 다음 명령을 사용하여 콘트롤 파일을 복원합니다.

```
restore controlfile from
'+DATA/orcl/controlfile/yourcontrolfilename';
그런 다음 데이터베이스를 마운트하고 엡니다.
```

- a) **oraenv**를 사용하여 **orcl** 데이터베이스에 대한 환경을 설정한 다음 **RMAN**에 연결합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ rman target /
Recovery Manager: Release 11.2.0.2.0 - Production on Sat Aug
22 03:27:29 2009
```

연습 16-4: 콘트롤 파일 손실 Recovery (계속)

```
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database (not started)
```

```
RMAN>
```

- b) 먼저 데이터베이스를 NOMOUNT 모드로 설정해야 합니다.

```
RMAN > startup nomount
```

```
Oracle instance started
```

Total System Global Area	577511424 bytes
Fixed Size	1337984 bytes
Variable Size	469763456 bytes
Database Buffers	100663296 bytes
Redo Buffers	5746688 bytes

```
RMAN>
```

- c) +DATA 디스크 그룹에 있는 기존 콘트롤 파일로부터 콘트롤 파일을 복원합니다.

주: 4단계에서 확인된 파일 이름을 사용합니다.

```
RMAN> restore controlfile from  
'+DATA/orcl/controlfile/current.260.695209463';
```

```
Starting restore at 22-AUG-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=23 device type=DISK

channel ORA_DISK_1: copied control file copy
output file
name='+DATA/orcl/controlfile/current.260.695209463'
output file name='+FRA/orcl/controlfile/current.256.695532623'
Finished restore at 22-AUG-09
```

```
RMAN>
```

연습 16-4: 콘트롤 파일 손실 Recovery (계속)

- d) 데이터베이스를 마운트합니다.

```
RMAN> alter database mount;  
database mounted
```

- e) 데이터베이스를 엽니다.

```
RMAN> alter database open;  
database open
```

- 6) 브라우저를 열고 Enterprise Manager에 **dba1** 유저로 로그인합니다. Database home 페이지에서 데이터베이스가 다시 작동되어 실행 중인지 확인합니다.
- 7) 모든 세션을 종료하고 모든 window를 닫습니다.

단원 17의 연습

현재 상황: 최근에 HR 스키마에 대한 여러 질문을 받았습니다. 일상 업무에 지장을 주지 않고 이 질문을 분석하기 위해 Data Pump Wizard를 사용하여 HR 스키마를 파일로 эксп포트하려고 합니다. эксп포트를 수행할 때 이 스키마를 어느 데이터베이스로 임포트할지는 확실치 않습니다.

최종적으로 orcl 데이터베이스가 관리자측에서 임포트를 승인하는 유일한 데이터베이스라는 사실을 알게 됩니다. 따라서 Data Pump Wizard로 임포트를 수행하면서 HR 스키마를 DBA1 스키마로 재매핑합니다.

이어서 SQL*Loader를 사용해야 하는 두 개의 데이터 로드 요청을 수신합니다.

Oracle Internal & Oracle Academy
Use Only

연습 17-1: Data Pump를 사용하여 데이터 이동

이 연습에서는 먼저 DBA1 유저에게 DATA_PUMP_DIR 디렉토리에 대한 액세스를 제공하기 필요한 권한을 부여합니다. 그런 다음 원하는 테이블을 DBA1 스키마로 임포트할 수 있도록 HR 스카마를 эксп포트합니다. 이 연습에서는 EMPLOYEES 테이블만 임포트합니다.

- 1) 먼저 DBA1 유저에게 DATA_PUMP_DIR 디렉토리에 대한 적합한 권한을 부여하고 이 연습에 필요한 유저 및 룰을 생성해야 합니다. 이 연습을 위해 환경을 구성하는 데 필요한 모든 단계를 수행하는 스크립트가 준비되어 있습니다.
 - a) labs 디렉토리에서 다음을 실행하여 DBA1 유저에게 DATA_PUMP_DIR 디렉토리에 대한 권한을 부여하고 환경에 대한 다른 구성은 수행하는 **lab_17_01_01.sql** 스크립트를 검토합니다.

```
$ cat lab_17_01_01.sql
```

- b) lab_17_01_01.sh 스크립트는 lab_17_01_01.sql 스크립트를 호출합니다. **lab_17_01_01.sh** 스크립트를 실행합니다.

```
$ ./lab_17_01_01.sh
Create HR_TEST tablespace, HR_TEST_ROLE role, the HR_TEST
users

Tablespace created.

User created.

Role created.

Grant succeeded.

Grant succeeded.

Table altered.

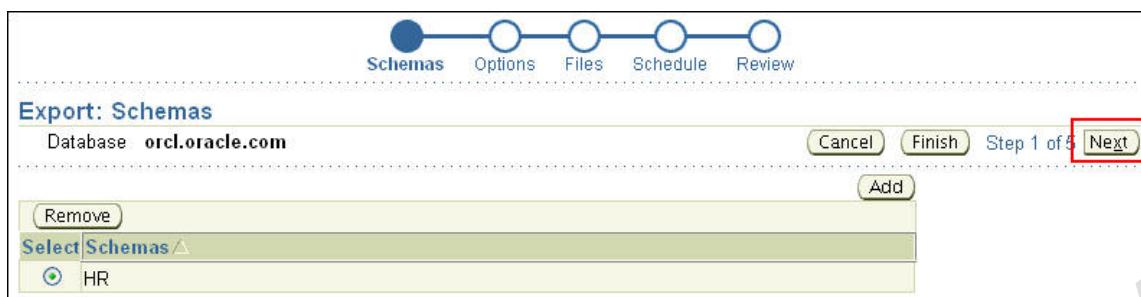
Grant succeeded.

Grant succeeded.

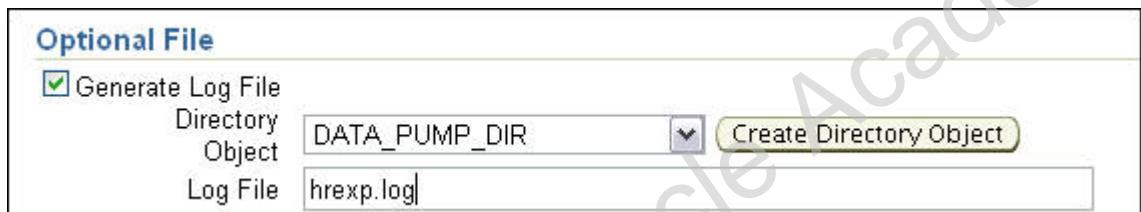
You may now login as: hr_test/oracle_4U
spool off
$
```

연습 17-1: Data Pump 를 사용하여 데이터 이동 (계속)

- 2) **Normal** 롤의 **DBA1** 유저로 Enterprise Manager에 로그인하여 HR 스키마를 엑스포트합니다.
- orcl** 데이터베이스에 대해 **Normal** 롤의 **DBA1** 유저로 Enterprise Manager를 호출합니다. **Connect As** 설정은 Normal이어야 합니다.
 - Data Movement > Move Row Data > Export to Export Files**를 선택합니다.
 - Schemas**를 선택하고 Username 및 Password에 **oracle**을 입력하고 **Save as Preferred Credential**을 선택한 다음 **Continue**를 누릅니다.
 - Export: Schemas 페이지에서 **Add**를 누르고 **HR** 스키마를 선택한 다음 **Select** 버튼을 누릅니다.
 - HR이 이제 스키마 리스트에 있는 것을 볼 수 있습니다. **Next**를 누릅니다.



- Export: Options 페이지의 Directory Objects drop-down list에서 **DATA_PUMP_DIR**을 선택하고 Log File에 **hrexp.log**를 입력합니다.



- Advanced Options를 검토하되 변경하지는 말고 **Next**를 누릅니다.
- Export: Files 페이지의 Directory Object drop-down list에서 **DATA_PUMP_DIR**을 선택하고 File Name에 **HREXP%U.DMP**를 입력한 다음 **Next**를 누릅니다.



연습 17-1: Data Pump 를 사용하여 데이터 이동 (계속)

- i) Export: Schedule 페이지에서 Job Name에 **hrexp**, Description에 **Export HR schema**를 입력하고 작업 시작 시간으로 Immediate를 적용한 다음 Next를 누릅니다.

Export: Schedule

Database **orcl.oracle.com**

Specify a name and description for the export job. Specify a date to start the job.

Job Parameters

Job Name **hrexp**
Description **Export HR schema**

Job Schedule

Time Zone **(UTC-06:00) US Central Time**

Start

Immediately
 Later

Next

- j) Export: Review 페이지에서 **Show PL/SQL**을 누르고 작업 생성을 지원하는 Export Wizard의 PL/SQL을 검토합니다.
- k) **Submit Job**을 눌러 작업을 제출합니다.
- l) HREXP 작업에 대한 링크를 눌러 진행률을 모니터합니다. 작업이 성공적으로 완료된 것을 확인할 때까지 브라우저를 refresh한 후 다음 단계로 이동합니다.

Summary	
Status	Succeeded
Scheduled	Jul 14, 2009 2:44:00 PM (UTC+07:00)
Started	Jul 14, 2009 2:44:00 PM (UTC+07:00)
Ended	Jul 14, 2009 2:45:45 PM (UTC+07:00)
Elapsed Time	105 seconds
Notification	No

- 3) 이제 엑스포트된 **HR** 스키마에서 **EMPLOYEES** 테이블을 **DBA1** 스키마로 임포트합니다. 명령행 인터페이스에 대한 감각을 익히기 위해 명령행에서 **impdp** 유ти리티를 사용하여 **EMPLOYEES** 테이블을 **DBA1** 유저 스키마로 임포트할 수 있습니다.
- a) **oraenv**를 실행하여 **orcl** 데이터베이스에 맞게 환경이 구성되었는지 확인합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

연습 17-1: Data Pump 를 사용하여 데이터 이동 (계속)

- b) 다음 전체 명령 문자열을 입력합니다. 명령 끝에 도달하기 전까지 [Enter]를 누르지 마십시오.

```
impdp dba1/oracle_4U DIRECTORY=data_pump_dir  
DUMPFILE=HREXP01.DMP REMAP_SCHEMA=hr:dba1  
TABLES=hr.employees LOGFILE=empimport.log
```

```
$ impdp dba1/oracle_4U DIRECTORY=data_pump_dir  
DUMPFILE=HREXP01.DMP REMAP_SCHEMA=hr:dba1  
TABLES=hr.employees LOGFILE=empimport.log

Import: Release 11.2.0.1.0 - Production on  
Thu Jul 16 00:46:18 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  
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  
Master table "DBA1"."SYS_IMPORT_TABLE_01" successfully  
loaded/unloaded  
Starting "DBA1"."SYS_IMPORT_TABLE_01": dba1/**********  
DIRECTORY=data_pump_dir DUMPFILE=HREXP01.DMP  
REMAP_SCHEMA=hr:dba1 TABLES=hr.employees  
LOGFILE=empimport.log  
Processing object type SCHEMA_EXPORT/TABLE/TABLE  
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA  
. . imported "DBA1"."EMPLOYEES"  
16.81 KB      107 rows  
Processing object type  
SCHEMA_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT  
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX  
Processing object type  
SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT  
Processing object type  
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS  
Processing object type SCHEMA_EXPORT/TABLE/COMMENT  
Processing object type  
SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT  
ORA-39083: Object type REF_CONSTRAINT failed to create with  
error:  
ORA-00942: table or view does not exist  
Failing sql is:  
ALTER TABLE "DBA1"."EMPLOYEES" ADD CONSTRAINT "EMP_JOB_FK"  
FOREIGN KEY ("JOB_ID") REFERENCES "DBA1"."JOBS" ("JOB_ID")  
ENABLE  
  
ORA-39083: Object type REF_CONSTRAINT failed to create with  
error:  
ORA-00942: table or view does not exist  
Failing sql is:
```

연습 17-1: Data Pump 를 사용하여 데이터 이동 (계속)

```
ALTER TABLE "DBA1"."EMPLOYEES" ADD CONSTRAINT "EMP_DEPT_FK"
FOREIGN KEY ("DEPARTMENT_ID") REFERENCES
"DBA1"."DEPARTMENTS" ("DEPARTMENT_ID") ENABLE

Processing object type SCHEMA_EXPORT/TABLE/TRIGGER
ORA-39082: Object type TRIGGER:"DBA1"."SECURE_EMPLOYEES"
created with compilation warnings
ORA-39082: Object type TRIGGER:"DBA1"."SECURE_EMPLOYEES"
created with compilation warnings
ORA-39082: Object type TRIGGER:"DBA1"."UPDATE_JOB_HISTORY"
created with compilation warnings
ORA-39082: Object type TRIGGER:"DBA1"."UPDATE_JOB_HISTORY"
created with compilation warnings
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS(TABLE_STATISTICS)
Job "DBA1"."SYS_IMPORT_TABLE_01" completed with 6 error(s)
at 00:46:45
```

주: EMPLOYEES 테이블만 임포트되고 스키마의 다른 객체는 임포트되지 않으므로 제약 조건 및 트리거가 생성되지 않는다는 오류가 나타날 수 있습니다. 이는 예상되는 오류입니다.

- c) 또한 로그 파일을 검토하여 임포트가 성공했는지 확인할 수 있습니다.

```
$ cat /u01/app/oracle/admin/orcl/dpdump/empimport.log
```

- 4) SQL*Plus에 **DBA1** 유저로 로그인하고 **EMPLOYEES** 테이블에서 데이터를 선택하여 EMPLOYEES 테이블이 DBA1 스키마에 로드되었는지 확인합니다.
- a) **DBA1** 유저로 SQL*Plus에 로그인합니다.
주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus dba1

SQL*Plus: Release 11.1.0.6.0 - Production on
Sun Aug 31 16:54:32 2008

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Enter Password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
Production
With the Partitioning, OLAP, Data Mining and Real
Application Testing options

SQL>
```

연습 17-1: Data Pump 를 사용하여 데이터 이동 (계속)

- b) 임포트가 되었는지 확인하기 위해 **DBA1** 스키마의 **EMPLOYEES** 테이블에서 행 수를 Select합니다.

```
SQL> select count(*) from employees;
```

```
 COUNT(*)  
-----  
      107  
SQL>
```

Oracle Internal & Oracle Academy
Use Only

연습 17-2: SQL*Loader를 사용하여 데이터 로드

이 연습에서는 SQL*Loader를 사용하여 Enterprise Manager Database Control을 통해 PRODUCT_MASTER 테이블에 데이터를 로드합니다. 데이터 및 콘트롤 파일이 제공됩니다.

- 1) **DBA1** 유저로 Enterprise Manager를 사용하여 **lab_17_02_01.dat** 데이터 파일을 로드합니다. 이 데이터 파일은 **PRODUCT_MASTER** 테이블에 대한 데이터 행을 포함합니다. **lab_17_02_01.ctl** 파일은 이 로드에 대한 콘트롤 파일입니다.
계속 진행하기 전에 선택적으로 lab_17_02_01.dat 및 lab_17_02_01.ctl 파일을 검토하여 해당 구조를 자세히 살펴볼 수 있습니다.
 - a) **orcl** 데이터베이스에 대해 **Normal** 롤의 **DBA1** 유저로 Enterprise Manager를 호출합니다.
 - b) **Data Movement > Move Row Data > Load Data from User Files**를 선택합니다.
 - c) **Use Existing Control File**을 누릅니다. 아직 입력하지 않은 경우 Username 및 Password에 각각 **oracle**을 입력하고 **Save as Preferred Credential**을 누른 다음 **Continue**를 누릅니다.

Load Data: Generate Or Use Existing Control File

Database **orcl.oracle.com** Cancel Continue

Automatically Generate Control File
A control file will be generated after you define the structure of the data file.

Use Existing Control File
Allows you to use an existing control file that defines the structure of the data file.

Host Credentials

* Username **oracle**
* Password *********
 Save as Preferred Credential

- d) Load Data: Control File 페이지에서 콘트롤 파일 이름 및 경로로 **/home/oracle/labs/lab_17_02_01.ctl**을 입력하거나 손전등 아이콘을 사용하여 이 콘트롤 파일을 선택합니다. **Next**를 누릅니다.

Load Data: Control File

Database **orcl.oracle.com** Cancel Finish Step 1 of 6 Next

A control file is used to describe what will be loaded and how. Specify the full path and name of the control file on the database server machine.

/home/oracle/labs/lab_17_02_01.ctl

연습 17-2: SQL*Loader 를 사용하여 데이터 로드 (계속)

- e) Load Data: Data File 페이지에서 **Provide the full path and name on the database server machine**을 누르고 데이터 파일 이름 및 경로로 **/home/oracle/labs/lab_17_02_01.dat**를 입력하거나 손전등 아이콘을 사용하여 이 데이터 파일을 선택합니다. Next를 누릅니다.

Load Data: Data File

Database **orcl.oracle.com** Step 2 of 6 **Next**

How would you like to specify the file containing the data?

The data file is specified in the control file
 Provide the full path and name on the database server machine
 /home/oracle/labs/lab_17_02_01.dat

- f) Load Data: Load Method 페이지에서 **Conventional Path**를 선택한 다음 Next를 누릅니다.

Load Data: Load Method

Database **orcl.oracle.com** Step 3 of 6 **Next**

Conventional Path
Runs SQL INSERT statements to load data into database tables. Use this option if none of the choices below are appropriate.

Direct Path (faster)
Formats data blocks and writes them directly to the database files. No writing to clustered tables. No other writing to destination tables in progress. No SQL in the control file.

Parallel Direct Path (fastest)
Writes data into the same table or into the same partition table in concurrent sessions. Data is only appended. Triggers and constraints are disabled. No indexes maintained.

- g) Load Data: Options 페이지에서 기본값을 모두 그대로 적용하되 로그 파일 이름과 경로로 **/home/oracle/labs/lab_17_02_01.log**를 입력합니다. 필요한 경우 Advanced Options를 검토하되 변경하지는 말고 Next를 누릅니다.

Optional Files

Generate bad file where records which contain errors are stored
Bad File

Generate discard file where rejected and uninserted records are stored
Discard File

Generate log file where logging information is to be stored
Log File /home/oracle/labs/lab_17_02_01.log

Show Advanced Options Step 4 of 6 **Next**

연습 17-2: SQL*Loader 를 사용하여 데이터 로드 (계속)

- h) Load Data: Schedule 페이지에서 Job Name으로 **lab_17_02_01**, Description으로 **Load data into the PRODUCT_MASTER table**을 입력합니다. 작업 시작 시간으로 Immediately를 선택한 다음 Next를 누릅니다.

The screenshot shows the 'Load Data: Schedule' page of an Oracle application. At the top, it says 'Database orcl.oracle.com'. Below that, it asks 'Specify a name and description for the load data job. Specify a date to start the job.' Under 'Job Parameters', the 'Job Name' is set to 'lab_17_02_01' and the 'Description' is 'Load data into the PRODUCT_MASTER table'. Both fields are highlighted with red boxes. In the 'Job Schedule' section, the 'Time Zone' is '(UTC-06:00) US Central Time'. Under 'Start', the radio button for 'Immediately' is selected and highlighted with a red box, while 'Later' is unselected. At the bottom right of the page, there are 'Cancel', 'Back', 'Step 5 of 6', and 'Next' buttons. The 'Next' button is highlighted with a red box.

- i) Load Data: Review 페이지에서 로드 정보와 파라미터를 검토한 다음 Submit Job을 누릅니다.
- j) LAB_17_02_01 작업에 대한 링크를 눌러 진행률을 모니터합니다. 작업이 성공적으로 완료된 것으로 표시된 후 다음 단계로 이동합니다.

The screenshot shows the 'Summary' page of the job execution. It displays the following information:

Status	Succeeded
Scheduled	Jul 14, 2009 2:55:29 PM (UTC+07:00)
Started	Jul 14, 2009 2:55:29 PM (UTC+07:00)
Ended	Jul 14, 2009 2:55:37 PM (UTC+07:00)
Elapsed Time	8 seconds
Notification	No

- k) /home/oracle/labs 디렉토리에 **lab_17_02_01.log** 파일을 검토하여 결과를 확인합니다.

연습 17-2: SQL*Loader 를 사용하여 데이터 로드 (계속)

- 2) **INVENTORY** 유저로 SQL*Loader 명령행을 사용하여 **PRODUCT_ON_HAND** 테이블에 데이터를 로드합니다. **lab_17_02_02.dat** 데이터 파일은 **PRODUCT_ON_HAND** 테이블에 대한 데이터 행을 포함합니다.

lab_17_02_02.ctl 파일은 이 로드에 대한 콘트롤 파일입니다.

계속 진행하기 전에 선택적으로 lab_17_02_02.dat 및 lab_17_02_02.ctl 파일을 검토하여 해당 구조를 자세히 살펴볼 수 있습니다.

- 터미널 window를 열고 **/home/oracle/labs** 디렉토리로 이동합니다.
- oraenv**를 실행하여 **orcl** 데이터베이스에 맞게 환경이 구성되었는지 확인합니다.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

- 다음 SQL*Loader 명령을 입력합니다(중간에 [Enter]를 누르지 말고 명령을 끝까지 입력).

```
sqlldr userid=inventory/oracle_4U
control=lab_17_02_02.ctl log=lab_17_02_02.log
data=lab_17_02_02.dat
```

```
$ sqlldr userid=inventory/oracle_4U control=lab_17_02_02.ctl
log=lab_17_02_02.log data=lab_17_02_02.dat

SQL*Loader: Release 11.2.0.1.0 - Production on
Tue Jul 14 14:56:44 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.
All rights reserved.

Commit point reached - logical record count 64
Commit point reached - logical record count 83
$
```

- /home/oracle/labs** 디렉토리에 **lab_17_02_02.log** 파일을 검토하여 결과를 확인합니다.

단원 18의 연습

Oracle Internal & Oracle Academy
Use Only

연습 18-1: Critical 오류 조사

이 연습에서는 Critical 오류를 조사합니다. 데이터 블록이 손상되었습니다. Enterprise Manager Support Workbench, Health Checker 및 Advisor를 사용하여 오류에 대응합니다.

- 1) 연습을 위해 데이터베이스를 설정합니다. 이 설정 작업에서는 연습을 위한 테이블스페이스를 생성하고 테이블스페이스 백업을 수행합니다.

lab_18_01_01.sh 스크립트를 실행하십시오. 이 연습에 필요한 스크립트는 **\$HOME/labs** 디렉토리에 있습니다.

```
$ cd $HOME/labs  
$ ./lab_18_01_01.sh
```

- 2) **sysdba**로 연결된 SQL*Plus에서 **lab_18_01_02.sql** 스크립트를 실행하여 Critical 오류를 생성합니다. 스크립트에서 블록 번호를 묻는 메시지를 표시하면 스크립트에 제공된 블록 번호를 입력합니다. 오류 메시지를 확인합니다. 이 시나리오에서는 오류 메시지가 원래 모든 유저에게 나타나고 DBA에게 즉시 보고되지 않을 수 있습니다.

주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
$ sqlplus / as sysdba
```

```
SQL*Plus: Release 11.2.0.1.0 Production on  
Wed Jul 15 01:30:42 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
```

```
SQL> @lab_18_01_02.sql  
Connected.
```

FILE_NO	BLOCK_NO
9	129

메시지가 표시되면 이
번호를 입력해야 하므로 이
블록 번호를 적어 두십시오.

```
System altered.
```

```
'Enter Block number when prompted'  
Enter value for block_no: 129  
0+1 records in  
0+1 records out  
80 bytes (80 B) copied, 6.693e-05 seconds, 1.2 MB/s
```

연습 18-1: Critical 오류 조사 (계속)

```
0+1 records in  
0+1 records out  
79 bytes (79 B) copied, 6.3419e-05 seconds, 1.2 MB/s  
  
SELECT * from js.jscopy  
*  
ERROR at line 1:  
ORA-01578: ORACLE data block corrupted (file # 9, block #  
129)  
ORA-01110: data file 9:  
'/u01/app/oracle/oradata/orcl/jstbs01.dbf'
```

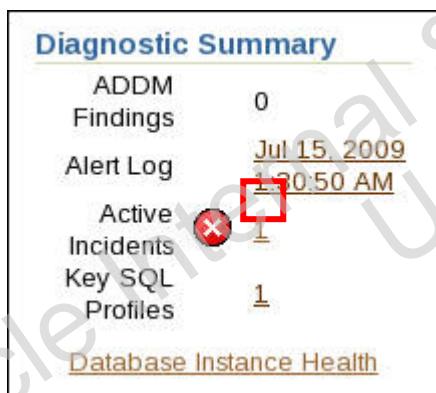
```
Disconnected from Oracle Database 11g Enterprise Edition  
Release 11.2.0.1.0 - Production  
With the Partitioning, Automatic Storage Management, OLAP,  
Data Mining  
and Real Application Testing options  
$
```

- 3) Enterprise Manager에 **SYSDBA** 률의 **DBA1** 유저로 로그인한 다음 Database home 페이지에서 alert를 확인하고 alert 세부 정보를 조사합니다. Alert에 Incident가 나타나면 Active Incidents 링크를 누릅니다.

하나 이상의 Critical Alert가 나타나야 합니다. 시간에 따라 다음과 같은 alert가 하나 이상 나타날 수 있습니다.

Severity	Category	Name	Impact	Message	Alert Triggered
✗	Data Failure	Data Failure Detected		Checker run found 2 new persistent data failures.	Jul 15, 2009 1:32:46 AM
✗	Incident	Oracle Data Block Corruption		An Oracle data block corruption detected in /u01/app/oracle/diag/rdbms/orcl/orcl/alert/log.xml at time/line number: Wed Jul 15 01:30:50 2009/18258.	Jul 15, 2009 1:32:46 AM
⚠	User Audit	Audited User		User SYS logged on from edtsr12p1.us.oracle.com.	Jul 15, 2009 1:17:22 AM

Active Incidents 수는 alert 수와 곧바로 일치하지 않을 수 있습니다. **Active Incidents** 링크를 누릅니다.



연습 18-1: Critical 오류 조사 (계속)

- 문제의 세부 정보를 조사합니다.

Support Workbench 페이지의 **Problems** 탭에서 ORA 1578 설명이 있는 문제를 확장합니다.

첫번째로 나열된 가장 최근 Incident의 Incident 번호를 누릅니다.

- Incident Details 페이지를 조사합니다. 오류를 수신한 명령문의 SQL 텍스트가 나열되어 있고 세션 정보가 나타납니다. 오류에 따라 이 정보는 오류 해결에 아주 유용할 수 있습니다.

- Trace File을 확인합니다.

안경 아이콘을 눌러 나열된 첫번째 Trace File을 봅니다. Host Login Credentials를 묻는 메시지가 나타나면 유저에 oracle, 암호에 oracle을 입력하고 Continue를 누릅니다.

주: 나열된 Trace File이 표시되지 않으면 잠시 동안 기다립니다. 리스트에 Trace File이 표시될 때까지 몇 분 정도 걸릴 수 있습니다.

Contents 페이지에서 Trace File을 볼 수 있습니다. Trace File의 색션은 확장하거나 축소할 수 있습니다. Trace File은 Oracle 내부 용도로 만들어졌습니다. **OK**를 누릅니다.

페이지 상단에 있는 경로 링크에서 **Support Workbench** 링크를 눌러 Support Workbench 페이지로 돌아갑니다.

- 문제 해결 방법을 결정합니다.

ORA 1578 문제의 Id를 누릅니다.

The screenshot shows a table with the following data:

	<input type="checkbox"/>	ID	Description	Number Of Incidents
	Hide	1	ORA 1578	1

연습 18-1: Critical 오류 조사 (계속)

- 8) 문제 세부 정보 페이지가 나타나면 Investigate and Resolve 섹션에서 스스로 해결 방법을 모색하거나 오라클 지원 센터에 도움을 신청할 수 있는 두 개의 탭을 볼 수 있습니다.

The screenshot shows the 'Investigate and Resolve' section of the Oracle Support interface. It includes tabs for 'Self Service' and 'Oracle Support'. Under 'Assess Damage', there are links for 'Checker Findings', 'Run Checkers', and 'Database Instance Health'. Under 'Diagnose', there are links for 'Alert Log', 'Related Problems Across Topology', 'Diagnostics for Last Dumped Incident', and 'Go to My Oracle Support and Research'. Under 'Resolve', there are links for 'SQL Repair Advisor' and 'Data Recovery Advisor'.

- 9) 문제에 대한 Checker 정보를 얻습니다. 페이지 상단에 있는 경로 링크에서 **Support Workbench**를 눌러 Support Workbench 페이지로 돌아갑니다. **Checker Findings** 탭을 누릅니다.

The screenshot shows the 'Support Workbench' page of the Oracle Support interface. The 'Checker Findings' tab is highlighted with a red box. Other tabs shown are 'Problems (1)' and 'Packages (0)'. Below the tabs is a search bar with fields for 'Description', 'Damage Translation', 'Status', and 'Time Detected'.

연습 18-1: Critical 오류 조사 (계속)

- 10) 권고를 얻습니다. Checker Findings 탭 아래의 Data Corruption 섹션에서 "Datafile ..."로 시작하는 설명이 있는 결과를 선택하고 **Launch Recovery Advisor**를 누릅니다.

Data Corruption						
Select findings and click on the "Launch Recovery Advisor" button to repair those findings.						
Launch Recovery Advisor						
Select All Select None Expand All Collapse All						
Select Description	Priority	Damage Translation	Incident ID	Status	Time Detected	
<input type="checkbox"/> ▾ All Findings						
<input checked="" type="checkbox"/> ▶ Datafile 9: '/u01/app/oracle/oradata/orcl/jstbs01.dbf' contains one or more corrupt blocks	High	Some objects in tablespace JSTBS might be unavailable	18345	Open	July 15, 2009 1:31:07 AM GMT+07:00	

주: 설명에 제공된 데이터 파일 번호는 2단계의 lab_18_01_02.sql 스크립트에 표시된 FILE_NO와 동일합니다.

- a) View and Manage Failures 페이지에서 Failure "Datafile ..."이 선택되어 있는지 확인한 다음 Advise를 누릅니다.
- b) Recovery Advice 페이지에서 RMAN 스크립트를 권장합니다. 이 스크립트는 Block Media Recovery를 수행합니다. Continue를 누릅니다.
- c) Review 페이지에 해결 할 Failure가 표시됩니다. Submit Recovery Job을 누릅니다.
- d) Processing 페이지가 짧게 나타날 수 있습니다. 이어서 Job Activity 페이지가 나타납니다. Job Status에 Running이 표시됩니다. 잠시 후 페이지를 Refresh하고 페이지에 No Jobs Found가 표시될 때까지 반복합니다.

연습 18-1: Critical 오류 조사 (계속)

- 11) 터미널 window를 열고 유저 **js** 및 암호 **oracle_4U**로 SQL*Plus에 로그인합니다. **jscopy** 테이블에서 선택하고 추가 블록 손상 오류가 표시되지 않는지 확인합니다.
- 주: 아직 터미널 window에서 데이터베이스에 환경을 설정하지 않은 경우 oraenv를 사용하여 환경을 orcl 데이터베이스에 맞게 설정합니다.

```
SQL> select * from jscopy;

EMPLOYEE_ID FIRST_NAME           LAST_NAME
-----  -----
EMAIL          PHONE_NUMBER        HIRE_DATE
JOB_ID         SALARY
-----  -----
-----  -----
COMMISSION_PCT MANAGER_ID        DEPARTMENT_ID
-----  -----
100  Steven            King
SKING          515.123.4567      17-JUN-87
AD_PRES        24001
                           90
                           ...
101  Neena             Kochhar
NKOCHHAR       515.123.4568      21-SEP-89
AD_VP          17001
                           100
                           90
                           ...
...
```

- 12) 완료되면 **\$HOME/labs** 디렉토리에서 **lab_18_01_12.sh** 스크립트를 실행하여 이 연습에서 생성된 객체를 제거합니다.

```
$ cd $HOME/labs
$ ./lab_18_01_12.sh
```

부록 B

기본 Linux 및

vi 명령

Oracle Internal & Oracle Academy
Use Only

vi 명령

Visual Interpreter/Editor(*vi*)는 UNIX 환경에서 사용할 수 있는 텍스트 편집기로 가장 널리 사용됩니다. *vi*는 어려운 명령 구문으로 인해 사용하기가 어렵지만, 거의 모든 버전의 UNIX 및 Linux 운영 체제에 포함되는 유일한 편집기입니다. 다음은 사용 가능한 *vi* 명령 리스트의 일부입니다.

*vi*에는 명령행 모드(입력하는 모든 내용을 편집 명령으로 간주)와 입력 모드(입력하는 모든 내용을 편집할 파일의 일부로 간주)의 두 가지 모드가 있습니다. 입력 모드로 들어가려면 a, A, i, I, o, O, c, C, s, S, r 또는 R을 입력하고, 명령행 모드로 돌아가려면 <ESC> 키를 사용합니다. SQLPlus에서 *vi* 편집기에 액세스하려면 다음 명령을 입력합니다.

```
SQL>define _editor=vi
```

파일을 편집하려면 SQLPlus 프롬프트에서는 edit <filename>을 입력한 다음 Enter를 누르고, Linux 명령 프롬프트에서는 vi <filename>을 입력한 다음 Enter를 누릅니다.

커서를 이동하려면 다음 옵션을 사용하십시오.

h - 원쪽으로 이동 j - 아래로 이동 k - 위로 이동 l - 오른쪽으로 이동

w - 한 단어 앞으로 b - 한 단어 뒤로 e - 현재 단어 끝으로

W, B 또는 E - 해당 소문자와 동일하지만 구두점 무시

0(영) - 현재 행의 처음으로 이동 \$ - 현재 행의 끝으로 이동

G - 파일의 마지막 행으로 이동 H - 화면의 맨 위 행으로 이동

L - 화면의 마지막 행으로 이동 M - 화면의 중간 행으로 이동

/<string> - <string>의 다음 항목 앞으로 검색

?<string> - <string>의 다음 항목 뒤로 검색

n - 이전 검색 반복

N - 이전 검색을 반대 방향으로 반복

<ctrl> f - 한 페이지 앞으로 이동

<ctrl> b - 한 페이지 뒤로 이동

이전 변경 사항을 언두하려면 다음 옵션을 사용하십시오.

u - 가장 최근 변경 사항을 언두

U - 가장 최근에 삭제한 텍스트를 언두

:e! - 마지막 변경 이후에 변경한 내용을 저장하지 않고 현재 파일을 다시 편집

새 텍스트를 입력하려면 다음 옵션을 사용하십시오.

a - 현재 커서 위치 뒤에 텍스트 추가

A - 행 끝에 텍스트 추가(행 끝으로 이동하여 추가 시작)

c - 객체 변경

C - 현재 커서 위치에서 행 끝까지 변경

i - 현재 커서 위치 앞에 텍스트 삽입

I - 행의 맨 앞에 텍스트 삽입

o - 현재 커서 위치 아래에 빈 행 삽입

O - 현재 커서 위치 위에 빈 행 삽입

r - 현재 커서 위치에 있는 문자 바꾸기

R - <ESC>를 누를 때까지 모든 문자 바꾸기

s - 커서 아래 문자를 텍스트로 대체

:s/A/B/opt 문자열 A를 문자열 B로 대체. %s/A/B/opt는 전역 바꾸기

옵션:

g(현재 행의 모든 항목 변경)

c(변경 전에 확인)

p(변경된 행 인쇄)

S - 전체 행 끝까지 대체

. <period> - 마지막 변경 내용 반복

n. <integer><period> 마지막
변경 내용 n회 반복

입력 모드를 끝내려면 <ESC>를 누릅니다.

기존 텍스트를 삭제하려면 다음 옵션을 사용하십시오.

x - 현재 커서 위치 바로 아래의 문자 삭제

dd - 현재 커서가 있는 전체 행 삭제

dnd(여기서 *n*은 정수) 현재 커서 위치에서 *n*개 행 삭제

dw - 현재 단어 삭제

D - 현재 행 끝까지 삭제

J - 현재 행 끝의 반환 삭제. 이 행을 다음 행과 연결

<int> J - 다음 <int>개 행 연결

복사, 잘라내기 및 붙여넣기: vi는 마지막으로 변경하거나 삭제한 텍스트가 저장되는 단일 버퍼를 사용합니다. 이 텍스트는 다음 명령을 사용하여 조작할 수 있습니다.

Y - 현재 행 복사

y <integer> - 다음 <int>개 행 복사

yw - 현재 단어 복사

yb - 앞 단어 복사

p - 커서 뒤에 버퍼 내용 넣기

P - 커서 앞에 버퍼 내용 넣기

입력 셋션의 s 및 S 명령을 참조하십시오.

편집한 변경 내용을 운영 체제 파일에 저장하려면 다음 옵션을 사용하십시오.

zz - 편집 모드 종료

:w filename - 변경 내용을 지정한 파일 이름으로 저장

:wq - 모든 변경 내용을 기록하고 편집 모드 종료

변경 내용을 저장하지 않고 종료하려면 다음 옵션을 사용하십시오.

ZZ - 편집 모드 종료

:q! - 변경 내용을 저장하지 않고 파일 종료

기본 Linux 명령

이 부록은 강의를 들을 때 신속하게 참조하는 용도로만 사용하십시오. 이 명령에 대한 자세한 내용은 온라인 설명서, Linux 설명서 또는 기타 Linux 명령 참고서를 참조하십시오.

파일 및 디렉토리	Linux 명령	설명/주석
명령 설명서	man <command> man -k <string> man man	o] <command>에 대한 설명서 항목을 찾습니다. o] <string>을 포함하는 설명서 항목을 모두 표시합니다. man에 대한 설명서 페이지를 표시합니다.
명령 정보	info <command>	이 명령에 대한 정보 시스템 항목을 표시합니다. info info를 사용하면 info 설명서 시스템의 자습서가 표시됩니다.
표준 출력으로 출력	cat <file>	연결 및 출력 – 명명된 파일을 터미널 화면에 출력합니다.
유저 나열	cat /etc/password	
작업 디렉토리 변경	cd <directory>	작업 디렉토리를 지정한 디렉토리로 변경합니다. cd에 파라미터를 지정하지 않으면 \$HOME으로 변경됩니다.
파일 복사	cp <source_file> <destination_file>	소스 파일을 대상 파일로 복사합니다.
파일 보기	less <file>	한 번에 한 페이지씩 파일을 봅니다. more 또는 pg의 GNU 버전입니다.
파일 보기	more <file>	한 번에 한 페이지씩 파일을 봅니다. BSD 버전입니다.
디렉토리 나열	ls <directory>	옵션 -l은 긴 리스트, -R은 반복, -a는 숨겨진 파일 표시, -t는 시간별 정렬, -r은 역방향 정렬을 나타냅니다. 기본 디렉토리는 현재 작업 디렉토리입니다.
디렉토리 생성	mkdir <directory>	디렉토리 기본값을 현재 작업 디렉토리로 설정하고 전체 경로를 지정할 수도 있습니다.
파일 이동 또는 이름 바꾸기	mv <old_file> <new_file>	파일의 이름을 변경하거나 파일을 다른 디렉토리로 이동합니다.

프로세스 리스트	<code>ps</code> <code>ps -ef</code>	프로세스 보고서를 표시합니다. 시스템의 모든 프로세스를 전체 리스트로 표시합니다. 추가 옵션에 대한 자세한 내용은 온라인 설명서를 참조하십시오.
작업 디렉토리 출력	<code>pwd</code>	현재 작업 디렉토리를 <code>stdout</code> 으로 출력합니다.
파일 제거 또는 지우기	<code>rm <file></code>	Linux에서 파일을 제거하면 영구적으로 제거됩니다. 옵션 <code>-r</code> 은 반복, <code>-f</code> 는 강제(서브 디렉토리 포함)를 나타내며 매우 위험합니다. <code>rm</code> 명령은 <code>rm -i</code> 의 alias입니다. 옵션 <code>-i</code> 를 사용하면 'Are you sure?'라는 메시지가 표시됩니다.
빈 파일 생성	<code>touch <file></code>	파일을 생성합니다.
시스템 이름	<code>hostname</code>	시스템 이름을 반환합니다.
시스템의 IP 주소	<code>host <machine_name></code>	도메인 이름 서버를 query하고 시스템 이름의 IP 주소를 반환합니다.
원격 셸	<code>rsh <host> <command></code>	<code><host></code> 에서 <code><command></code> 를 실행합니다. Rsh는 안전하지 않으므로 ssh를 대신 사용합니다.
원격 셸	<code>ssh <host></code>	보안 셸에는 rsh, rcp, ftp 및 telnet을 대체하는 기능이 있습니다.
원격 셸	<code>telnet <host></code>	<code><host></code> 에서 터미널 세션을 시작합니다. Telnet은 안전하지 않으므로 ssh를 대신 사용합니다.
파일에서 패턴 검색	<code>grep <option> <pattern> <file></code>	<code><file></code> 또는 스트림에서 <code><pattern></code> 으로 정의되는 정규식을 검색하고 해당 패턴이 포함된 행을 표시합니다. 공통 옵션은 <code>-i</code> 로 대소문자 구분 없이 사용됩니다. grep는 다음과 같이 파이프를 통해 파일 또는 <code>stdin</code> 으로부터 입력을 받아들일 수 있습니다. <code>netstat -a grep ESTABLISHED</code>
스크립트 소스 지정	<code>. <script_file></code>	bash 셸에서 이 '!' 명령은 스크립트가 셸에서 강제로 실행되도록 합니다. 일반 동작은 스크립트가 하위 셸에서 실행되는 것입니다.

인터프리터	awk	입력을 해석하거나 형식을 다시 지정하는 매크로 언어입니다. 입력의 각 행에 대해 다양한 작업을 수행할 수 있습니다. ?new awk?를 nawk라고 할 수 있습니다.
파일 정렬	sort	stdin 또는 파일 이름 인수로부터 입력을 가져와 여러 옵션을 사용하여 특정 열, 필드 등을 기준으로 정렬합니다. 온라인 설명서를 참조하십시오.
명령행 편집기	sed	Sed는 명령행 편집기이며 셸 스크립트 편집에 유용한 많은 명령과 옵션을 사용할 수 있습니다.
Visual Editor	vi <file>	모든 Unix 시스템에서 사용할 수 있는 터미널 기반 편집기입니다. Linux에서는 vi의 대체로 개선된 vi인 vim을 제공합니다.
Gnu 편집기	emacs <file>	포괄적인 커스터마이즈가 가능한 기능을 갖춘 GPL 편집기로 대부분의 UNIX와 Linux 배포에서 사용할 수 있습니다.
WSIWIG 편집기	gedit <file>	전체 화면 편집기로 X가 있어야 합니다. Gnome에서 사용할 수 있습니다.
WSIWIG	kate <file>	전체 화면 편집기로 X가 있어야 합니다. KDE에서 사용할 수 있습니다.
터미널 출력	stdout	표준 출력(stdout)으로, 명령이 아니라 개념입니다. 대부분의 Linux 명령은 재지정되지 않는 한 기본적으로 stdout에 기록합니다.
터미널 입력(키보드)	stdin	표준 입력(stdin)으로, 명령이 아니라 개념입니다. 대부분의 Linux 명령은 재지정되지 않는 한 기본적으로 stdin에서 읽어옵니다.
Alias	alias <command> <alias>	유저가 <command>를 입력할 때 대체를 만들어 <alias>를 대체하고 실행합니다. 공통 alias는 alias 'rm' 'rm -i'입니다. 이 alias는 .bashrc 파일에서 설정합니다.
셸 변수 표시	set	현재 셸에 정의되어 있는 모든 변수를 출력합니다.

환경 변수 표시	printenv or env	하위 프로세스에서 상속할 수 있도록 '엑스포트'된 모든 환경 변수를 출력합니다.
파일 생성 마스크	umask -S u=rwx,g=rx,o=rx	이 셸 또는 하위 셸에서 생성한 모든 파일에 대해 기본 사용 권한을 설정합니다. -S 옵션은 기호 표기법을 사용하고 숫자 표기법은 더 이상 사용되지 않습니다.
시계	xclock	화면에 시계를 표시하는 X 클라이언트입니다. X windows 시스템을 테스트하는 데 자주 사용됩니다.
X 액세스 제어	xhost xhost +<Xclient>	현재 액세스 제어를 표시합니다. 로컬 DISPLAY에 액세스할 수 있는 Xclient를 추가합니다. <Xclient>가 지정되지 않은 경우에는 모두 허용됩니다.

Oracle Internal & Oracle Academy
Use Only

시스템 관리	Linux 명령	설명/주석
유저 대체	su - username	현재 작업을 수행 중인 유저를 변경합니다. 이 기능은 모든 유저가 사용하여 세션 유저의 유효 ID인 유저로 변경할 수 있습니다. 일반 유저는 암호를 제공해야만 하지만 루트 유저는 암호를 제공할 필요가 없습니다. '-' 파라미터는 선택 사항입니다. 이 파라미터는 새 유저 로그인 스크립트를 실행합니다.
제한된 루트 권한	sudo	루트 유저는 특정 명령을 루트로 실행할 수 있는 유저를 구성하고 암호가 필요한지 여부를 구성할 수 있습니다. 이 기능은 특정 유저가 특정 루트 명령을 수행하도록 하는 데에 유용합니다. 예를 들어, CDROM과 같이 제거 가능한 볼륨을 마운트 및 마운트 해제할 수 있습니다.
루트 파일 시스템	/	시스템 디렉토리 트리의 루트 디렉토리입니다.
홈 디렉토리	/home	일반적으로 모든 유저 홈 디렉토리가 있는 디렉토리입니다. 예제: /home/oracle
Tmp 디렉토리	/tmp	임시 저장 영역입니다. 보관할 파일을 여기에 저장하지 마십시오. SA에서는 크론 작업이 정기적으로 모든 파일을 제거하기도 합니다.
부트 디렉토리	/boot	커널 이미지와 부트 로더 명령을 저장하는 작은 Partition입니다.
로그 디렉토리	/var/log	대부분의 시스템 로그 파일이 있는 위치입니다.
예제 구성 파일	/etc/inittab	응용 프로그램별 구성 파일이 배포됩니다. 설치 후 변경한 모든 구성 파일은 백업에 포함되어야 합니다.

Password File	/etc/passwd /etc/shadow	/etc/passwd 파일은 유저 정보를 저장하며 다른 유저가 읽을 수 있어야 합니다. 암호화된 암호(encrypted password)를 사용하는 경우에도 보안 허점이 있을 수 있습니다. /etc/shadow 파일은 암호화된 암호를 저장하며 루트에서만 읽을 수 있습니다.
그룹 파일	/etc/group	/etc/groups 파일은 서버의 그룹과 그룹의 멤버인 유저를 정의합니다. 유저의 기본 그룹은 /etc/passwd 파일에 정의되어 있습니다.
X 구성 파일	/etc/X11/XF86Config	비디오 카드, 모니터, 마우스 및 키보드에 대한 X 서버 설정을 설정하는 파일입니다. 일반적으로 OS 공급업체에서 제공하는 도구로 설정됩니다.

일정이 잡힌 시간에 정기적으로 실행할 명령 일정 잡기	crontab -e	이 명령을 사용하여 crontab 파일을 편집하고 사용할 cron daemon의 사양을 생성합니다.
특정 빈도로 실행할 스크립트 일정 잡기	/etc/anacrontab	파일을 편집하여 특정 빈도로 실행할 스크립트를 지정합니다. (자세한 내용은 man anacrontab을 참조하십시오.)
지정한 시간에 한 번 실행할 명령 일정 잡기	at <options> TIME	<options>에서 지정한 작업을 지정한 TIME 파라미터에 실행합니다.
명령 일정 잡기	batch <options> <TIME>	로드 평균이 .8 이하로 내려가면 설정한 TIME 이후에 선택적으로 명령을 실행합니다.
파일 시스템 마운트	mount <opt> <dev> <mount_point>	<dev>에서 지정한 옵션을 사용하여 <mount_point>에서 <dev> 장치의 파일 시스템을 마운트합니다.
파일 시스템 마운트 해제	umount <dev> umount <mount_point>	파일 시스템 또는 장치의 마운트를 해제합니다.
유저 ID 최대 수	65535	
루트 암호 recovery	{lilo} control-x linux S passwd root {grub} c kernel vmlinuz-2.4.9-13 single ro root=/dev/hda8 initrd /initrd-2.4.9-13.img boot passwd root	손상된 루트 암호를 recovery하는 절차입니다. 이를 위해서는 시스템과 시스템 콘솔에 물리적으로 액세스해야 합니다. 시스템을 재부팅하여 시작한 다음 LILO 부팅 중에 [Ctrl] + [x]를 눌러 프롬프트를 표시하고 LILO에서 실행 레벨 S로 Linux를 부팅하도록 명령합니다. 두번째 프로시저에서는 grub 부트 로더를 사용합니다.
새 유저 생성	useradd	-D 옵션만 지정하면 기본값을 표시합니다. -D를 다른 옵션과 함께 지정하면 기본값 옵션이 변경되고, -D를 재정의하지 않으면 기본값(예: -g)이 기본 그룹을 설정합니다.

유저 삭제	userdel	유저를 제거하고 선택적으로 유저에 속한 모든 파일을 제거합니다.
유저 계정 수정	usermod	/etc/password 정보를 변경합니다.
새 그룹 생성	groupadd	-g는 그룹 ID를 설정합니다. 기본값은 사용 가능한 값 중 500보다 큰 첫번째 값입니다.
그룹 삭제	groupdel	시스템에서 그룹을 제거합니다. 유저의 기본 그룹인 그룹은 제거할 수 없습니다. 삭제한 그룹에서 소유한 파일은 chown을 사용하여 수동으로 변경해야 합니다.
실행 레벨 변경	init <runlevel>	init 명령은 rcN.d 스크립트를 평가하여 실행 레벨을 변경합니다. init 6을 사용하면 강제로 재부팅됩니다.
디스크 동기화	sync	버퍼 캐시와 페이지 캐시에서 모든 더티 버퍼를 디스크에 강제로 기록하도록 합니다. 디스크 손상을 방지하기 위해 재부팅 직전에 사용됩니다.
Linux 시스템 종료	shutdown <mode> <delay>	시스템을 정상적으로 종료하고 프로세스를 종료하며 모든 종료 스크립트를 실행하고 디스크를 동기화합니다. 모드는 -r(재부팅)과 -h(중지)가 있습니다. 지연(delay)은 필수 파라미터이며 숫자(초 단위) 또는 'now'를 지정합니다. 선택적으로 종료 경고 메시지가 표시될 수도 있습니다.
오류 로그	dmesg	부팅 메시지를 표시합니다. 이 로그는 순환적이며 일정 시간이 지나면 제한된 시스템 오류가 부팅 정보를 겹쳐쓸 수 있습니다.
네트워크 IP 구성	/etc/sysconfig/network-scripts/	이 디렉토리는 부트 시퀀스의 일부로 rc.sysinit에서 실행한 스크립트를 저장합니다.
호스트 IP 주소	/etc/hosts	유저의 시스템에서 인식하고 있는 호스트 리스트입니다. 적어도 로컬 시스템 이름과 루프백 IP가 포함되어야 합니다.
이름 서비스 스위치	/etc/nsswitch.conf	

네트워크 파라미터	sysctl -a grep net	커널에 설정된 모든 네트워크 파라미터를 표시합니다.
daemon 라우팅	routed	
NIC 구성	ifconfig -a	현재 구성되어 있는 모든 네트워크 장치를 표시합니다.
보조 IP 주소	modprobe ip_alias	
	ifconfig eth0:1 IP	
로그인 프롬프트	/etc/issue	로그인 프롬프트가 실행될 때 표시되는 배너 메시지입니다.
YP/NIS 서비스 바인더	/sbin/ypbind	이름 분석 및 기타 서비스를 위해 NIS 서버를 검색하여 연결합니다.
모듈 정보	modinfo <options> <module>	커널 모듈에 대한 정보를 표시합니다. -l은 라이센스, -p는 파라미터, -d는 설명을 표시합니다.
모듈 나열	lsmod	현재 로드된 모듈을 표시합니다.
모듈 로드	insmod	로드 가능 모듈을 로드합니다.
모듈 로드 취소	rmmmod	로드 가능 모듈의 로드를 취소합니다.
소프트웨어 설치	rpm -ivh package	-i는 설치, -v는 verbose, -h는 진행 상황 해시 표시를 나타냅니다.
소프트웨어 설치 해제	rpm -e package	-e는 패키지 지우기를 나타냅니다. 종속성이 있는 경우에는 소프트웨어가 설치 해제되지 않습니다.
설치된 소프트웨어 나열	rpm -qa	-q는 Query, -a는 모두를 나타냅니다. 설치된 패키지를 모두 나열합니다.
설치된 소프트웨어 확인	rpm -V package	설치된 파일을 rpm 데이터베이스 정보와 비교합니다.
모든 파일 나열	rpm -ql package	패키지에 속한 파일을 모두 나열합니다.
패키지 소유자	rpm -qf file	전체 파일 이름을 지정하는 경우 패키지를 나열합니다.
시스템 모델	uname -m	CPU 레벨(예: i686)을 표시합니다.
OS 레벨	uname -r	커널 버전을 표시합니다.
실행 레벨	runlevel	이전 및 현재 실행 레벨을 표시합니다.
커널 파라미터	sysctl -a	설정 가능한 모든 커널 파라미터의 설정을 표시합니다.

파일 기술자 최대 수	<code>sysctl fs.file-max</code>	프로세스당 최대 파일 기술자 수를 표시합니다.
커널 파라미터 설정	<code>/etc/sysctl.conf</code>	커널 파라미터에서 컴파일되며 부트 시 이 파일에서 설정하여 재설정할 수 있습니다.
커널 파라미터 변경	<code>echo <value> > /proc/<file></code>	커널 파라미터의 새 값을 /proc 파일 시스템에 씁니다.
	<code>echo 2147483648 >/proc/sys/kernel/shmmmax</code>	공유 메모리 세그먼트의 최대 크기 값을 설정합니다.
공유 메모리	<code>sysctl kernel.shmmmax</code>	shmmmax 파라미터를 표시합니다.
커널 파라미터 변경	<code>sysctl -w <parameter>=<value></code>	커널 파라미터를 변경합니다. -p 옵션은 파일의 설정을 읽어와 커널 파라미터를 설정합니다. 기본 파일은 <code>/etc/sysctl.conf</code> 입니다.
프로세스 제한 설정	<code>ulimit <option> <value></code>	셸 및 셸에서 시작한 프로세스에 대한 제한을 설정합니다. 유저가 보다 제한적으로 제한을 설정할 수 있으며 일반적으로 루트만 덜 제한적인 제한을 설정할 수 있습니다. 일부 옵션을 사용하려면 루트 권한이 있어야 합니다. 옵션: -u는 프로세스 수를 설정하고 -n은 파일 처리 수를 설정합니다. 기타 옵션은 man bash를 참조하십시오.
프로세스 제한 표시	<code>ulimit</code>	옵션을 사용하지 않는 경우 ulimit는 현재 제한 설정을 표시합니다.
프로세스 간 통신 (공유 메모리 및 세마포어)	<code>ipcs <option></code>	옵션: -m은 공유 메모리의 현재 사용량, -s는 세마포어 사용량, -a는 모두를 표시합니다.
공유 메모리 세그먼트 제거	<code>ipcrm shm <shmid></code>	<shmid>에서 확인한 공유 메모리 세그먼트를 해제합니다. 이 명령은 매우 위험합니다. 해제되는 세그먼트를 사용하는 데이터베이스가 손상될 수 있습니다.

시스템 성능	Linux 명령	설명/주석
성능 모니터	top	OS 및 프로세스 통계를 실시간으로 표시합니다.
시스템 동작 보고	sar <options> <interval> <count>	옵션: -q는 CPU 큐 표시, -u는 CPU 활용, -d는 장치 작업, -n은 DEV 네트워크 장치 작업 등(man 페이지 참조)이 있습니다. 간격은 초 단위입니다.
가상 메모리 통계	vmstat <interval> < count>	간격은 초 단위입니다.
가상 메모리 통계	cat /proc/meminfo	동시 가상 메모리 사용량을 표시합니다.
커널 캐시 통계	cat /proc/slabinfo	커널 슬랩 할당자 통계: inode, dentries 및 비동기 IO 버퍼와 같은 자주 할당되는 캐시 객체입니다.
I/O 통계	iostat <option> <interval> <count>	옵션: -d는 장치 작업, -c는 CPU 작업, -x는 확장 디스크 작업 통계를 나타냅니다. 간격은 초 단위입니다.
다중 처리기 통계	mpstat -P <cpu> <count> <interval>	smp 시스템의 특정 프로세서 또는 모든 CPU에 대한 CPU 통계를 반환합니다.
물리적 RAM	64 GB (Theoretical)	엔터프라이즈 커널에서 요구하는 최대 물리적 RAM입니다. Red Hat Enterprise Linux AS 21은 최대 16GB까지만 지원합니다.
교체 장치	swapon -s	교체용으로 사용 중인 장치를 표시합니다. 교체 장치는 설치 시 임의로 지정되며 변경하거나 추가할 수 있습니다. 여러 교체 장치를 생성할 수 있으며, 교체 장치 크기는 적어도 물리적 메모리보다 커야 합니다.

교체 장치 크기 표시	<code>free</code>	현재 메모리와 교체 장치 사용량을 표시합니다.
교체 장치 활성화	<code>swapon -a</code>	교체 장치를 활성화합니다.
사용 가능한 디스크 블록	<code>df -k</code>	KB 단위로 측정되며 MB 단위의 경우 <code>-m</code> 을 사용합니다.
장치 나열	<code>cat /proc/devices</code>	주요 및 보조 번호를 기준으로 시스템이 인식한 장치를 나열합니다.
디스크 정보	<code>Cat /proc/scsi/scsi0/sda/model</code> <code>Cat /proc/ide/ide0/hda/model</code>	SCSI 디스크 정보를 표시합니다. IDE 디스크 정보를 표시합니다.
네트워크 통계 출력	<code>netstat <options></code>	광범위한 네트워크 통계를 출력합니다(<code>man netstat</code> 를 참조).
그래픽 시스템 통계 뷰어	<code>xosview</code>	현재 OS 통계를 X 기반으로 표시합니다.

Oracle Internal & Oracle Academy
Use Only

기타 시스템 정보	Linux 명령	설명/주석
NFS 엑스포트	/etc(exports	데이터베이스 파일은 단순한 NFS에서 지원되지 않습니다.
NFS 클라이언트 마운트 디렉토리	/var/lib/nfs/xtab	
최대 파일 시스템	2 TB with 4KB block size (on 32 kernel)	ext3 및 ext2를 사용하는 다른 파일 시스템의 경우 다를 수 있습니다.
최대 파일 크기 파일 크기는 파일 시스템 크기를 초과할 수 없음	2 GB {512B block size} 2 TB {4KB block size}	오라클 데이터베이스에서는 데이터베이스 블록 크기가 16KB인 최대 64GB의 파일을 생성할 수 있습니다. 32비트 커널은 파일과 블록 장치 크기를 2TB로 제한합니다.
파일 시스템 블록 크기	dump2fs <device>	파일 시스템 속성을 stdout에 덤프합니다.
파일 시스템 테이블	/etc/fstab	부트 시 이 파일 시스템을 마운트합니다.
저널 파일 시스템 유형	ext3 reiserfs	
디스크 레이블	fdisk -l	일부 배포에서는 fdisk를 사용할 수 없습니다.
파일 시스템 확장	resize2fs resize_reiserfs	파일 시스템 확장은 일부 파일 시스템 유형에만 적용할 수 있습니다.
백업	tar cvf /dev/rst0 /	루트/파일 시스템의 백업을 생성합니다.
복원	tar xvf /dev/rst0	루트/파일 시스템을 복원합니다.
부트 볼륨 준비	/sbin/lilo	/etc/lilo.conf를 변경한 후 변경 내용을 부트 로더에 적용하려면 이 명령을 실행해야 합니다.
시작 스크립트	/etc/rc.d/rc	
커널	/boot/vmlinuz	
커널 비트	getconf WORD_BIT	커널 정보를 얻기 위한 POSIX 호출입니다. WORD_BIT 외에도 다른 변수가 많이 있습니다.

단일 유저 부트	<pre>{lilo} control-x linux S {grub} c kernel vmlinuz-2.4.9- 13 single ro root=/dev/hda8 initrd /initrd-2.4.9- 13.img boot</pre>	LILO 기능을 사용합니다. GRUB 부트 로더를 사용합니다.
시간대 관리	/etc/sysconfig/clock	
SW 디렉토리	/var/lib/rpm	rpm 데이터베이스가 저장되는 디렉토리입니다.
장치	/dev	모든 장치 파일을 보관하는 디렉토리입니다.
CPU	cat /proc/cpuinfo	CPU 통계 정보를 표시합니다.
전체 디스크	/dev/sda	장치 이름입니다.
CDROM	/dev/cdrom	일반적으로 /mnt/cdrom에서 마운트됩니다.
CDROM 파일 유형	iso9660	
플로피 드라이브	/dev/fd0	일반적으로 /mnt/floppy에서 마운트됩니다.
시스템 정보	/proc	/proc 파일 시스템은 프로세스 및 커널 설정과 통계에 액세스할 수 있는 메모리 기반 파일 시스템입니다.
실행 파일 컴파일 및 링크	make -f <file> <command>	대형 프로그램에서 재컴파일 할 부분을 결정하고 사용할 실행 파일을 컴파일, 링크 및 준비하는데 필요한 명령을 실행하려면 make 파일 <file>을 사용합니다.

LVM	Linux(UnitedLinux)	설명/주석
LVM	Logical Volume Manager	이 패키지는 Red Hat Enterprise Linux AS 2.1에서 제공되지 않으면 커널을 손상시키지 않고 추가할 수 없습니다. 커널 지원은 United Linux에서 제공됩니다.
LVM 개념	logical extents	논리 볼륨은 논리 extent로 구성됩니다.
	logical volume	일련의 논리 extent는 볼륨 그룹에서 가져오며 OS에는 디스크 볼륨으로 표시됩니다. 이러한 extent는 여러 디스크에서 스트립할 수 있습니다.
	volume group	fdisk 등으로 생성하고 pvcreate로 초기화한 다음 vgcreate를 사용하여 물리적 볼륨으로 그룹화한 물리적 디스크 partition 집합입니다.
볼륨 그룹 표시	vgdisplay -v	
물리적 볼륨 수정	pvchange	
물리적 디스크 준비	pvcreate	
물리적 볼륨 나열	pvdisplay	
볼륨 그룹에서 디스크 제거	vgreduce	
논리적 볼륨을 다른 물리적 볼륨으로 이동	pvmove	
볼륨 그룹 생성	vgcreate	
볼륨 그룹 제거	vgremove	
볼륨 그룹 가용성	vgchange	
볼륨 그룹 복원	vgcfgrestore	

볼륨 그룹 엑스포트	<code>vgexport</code>	
볼륨 그룹 임포트	<code>vgimport</code>	
볼륨 그룹 나열	<code>vgscan</code>	
논리 볼륨 특성 변경	<code>lvchange</code>	
논리 볼륨 나열	<code>lvdisplay</code>	
논리 볼륨 만들기	<code>lvcreate</code>	
논리 볼륨 확장	<code>lvextend</code>	
논리 볼륨 축소	<code>lvreduce</code>	
논리 볼륨 제거	<code>lvremove</code>	
스트라이핑된 볼륨 생성	<code>lvcreate -i 3 -I 64</code>	

Oracle Internal & Oracle Academy
Use Only

부록 C

SQL 문 구문

Oracle Internal & Oracle Academy
Use Only

SQL 문

Oracle Database SQL Quick Reference 설명서에서 발췌한 이 내용은 Oracle SQL 문의 구문을 보여줍니다. SQL 문은 프로그램 및 유저가 오라클 데이터베이스의 데이터에 액세스할 때 사용하는 수단입니다.

표 1은 각 SQL 문 및 관련 구문을 보여 줍니다. 표 2는 표 1에 있는 하위 절의 구문을 보여 줍니다.

참조: Oracle Database SQL Reference에서 Oracle SQL에 대한 자세한 내용을 확인할 수 있습니다.

표 1: SQL 문의 구문

SQL 문	구문
ALTER CLUSTER	<pre>ALTER CLUSTER [schema.]cluster { physical_attributes_clause SIZE size_clause allocate_extent_clause deallocate_unused_clause { CACHE NOCACHE } } [physical_attributes_clause SIZE size_clause allocate_extent_clause deallocate_unused_clause { CACHE NOCACHE }]... [parallel_clause] ;</pre>
ALTER DATABASE	<pre>ALTER DATABASE [database] { startup_clauses recovery_clauses database_file_clauses logfile_clauses controlfile_clauses standby_database_clauses default_settings_clauses redo_thread_clauses security_clause } ;</pre>

SQL 문	구문
ALTER DIMENSION	<pre> ALTER DIMENSION [schema.]dimension { ADD { level_clause hierarchy_clause attribute_clause extended_attribute_clause } [ADD { level_clause hierarchy_clause attribute_clause extended_attribute_clause }]... DROP { LEVEL level [RESTRICT CASCADE] HIERARCHY hierarchy ATTRIBUTE attribute [LEVEL level [COLUMN column [, COLUMN column]...] } [DROP { LEVEL level [RESTRICT CASCADE] HIERARCHY hierarchy ATTRIBUTE attribute [LEVEL level [COLUMN column [, COLUMN column]...] }]... COMPILE } ; </pre>
ALTER DISKGROUP	<pre> ALTER DISKGROUP { disk_clauses diskgroup_clauses } [{ disk_clauses diskgroup_clauses }]... ; </pre>
ALTER FUNCTION	<pre> ALTER FUNCTION [schema.]function COMPILE [DEBUG] [compiler_parameters_clause [compiler_parameters_clause] ...] [REUSE SETTINGS] ; </pre>

SQL 문	구문
ALTER INDEX	<pre> ALTER INDEX [schema.]index { deallocate_unused_clause allocate_extent_clause shrink_clause parallel_clause physical_attributes_clause logging_clause } [deallocate_unused_clause allocate_extent_clause shrink_clause parallel_clause physical_attributes_clause logging_clause]... rebuild_clause PARAMETERS ('ODCI_parameters') { ENABLE DISABLE } UNUSABLE RENAME TO new_name COALESCE { MONITORING NOMONITORING } USAGE UPDATE BLOCK REFERENCES alter_index_partitioning } ; </pre>
ALTER INDEXTYPE	<pre> ALTER INDEXTYPE [schema.]indextype { { ADD DROP } [schema.]operator (parameter_types) [, { ADD DROP } [schema.]operator (parameter_types)]... [using_type_clause] COMPILE } ; </pre>
ALTER JAVA	<pre> ALTER JAVA { SOURCE CLASS } [schema.]object_name [RESOLVER ((match_string [,] { schema_name - }) [(match_string [,] { schema_name - })]...)] { { COMPILE RESOLVE } invoker_rights_clause } ; </pre>

SQL 문	구문
ALTER MATERIALIZED VIEW	<pre> ALTER MATERIALIZED VIEW [schema.](materialized_view) [physical_attributes_clause table_compression LOB_storage_clause [, LOB_storage_clause]... modify_LOB_storage_clause [, modify_LOB_storage_clause]... alter_table_partitioning parallel_clause logging_clause allocate_extent_clause shrink_clause { CACHE NOCACHE }] [alter_iot_clauses] [USING INDEX physical_attributes_clause] [MODIFY scoped_table_ref_constraint alter_mv_refresh] [{ ENABLE DISABLE } QUERY REWRITE COMPILE CONSIDER FRESH] ; </pre>
ALTER MATERIALIZED VIEW LOG	<pre> ALTER MATERIALIZED VIEW LOG [FORCE] ON [schema.]table [physical_attributes_clause alter_table_partitioning parallel_clause logging_clause allocate_extent_clause shrink_clause { CACHE NOCACHE }] [ADD { { OBJECT ID PRIMARY KEY ROWID SEQUENCE } [(column [, column]...)] (column [, column]...) } [, { { OBJECT ID PRIMARY KEY ROWID SEQUENCE } [(column [, column]...)] (column [, column]...) }] ... [new_values_clause]] ; </pre>

SQL 문	구문
ALTER OPERATOR	<pre>ALTER OPERATOR [schema.]operator { add_binding_clause drop_binding_clause COMPILE } ;</pre>
ALTER OUTLINE	<pre>ALTER OUTLINE [PUBLIC PRIVATE] outline { REBUILD RENAME TO new_outline_name CHANGE CATEGORY TO new_category_name { ENABLE DISABLE } } [REBUILD RENAME TO new_outline_name CHANGE CATEGORY TO new_category_name { ENABLE DISABLE }]... ;</pre>
ALTER PACKAGE	<pre>ALTER PACKAGE [schema.]package COMPILE [DEBUG] [PACKAGE SPECIFICATION BODY] [compiler_parameters_clause [compiler_parameters_clause] ...] [REUSE SETTINGS] ;</pre>
ALTER PROCEDURE	<pre>ALTER PROCEDURE [schema.]procedure COMPILE [DEBUG] [compiler_parameters_clause [compiler_parameters_clause] ...] [REUSE SETTINGS] ;</pre>
ALTER PROFILE	<pre>ALTER PROFILE profile LIMIT { resource_parameters password_parameters } [resource_parameters password_parameters]... ;</pre>
ALTER RESOURCE COST	<pre>ALTER RESOURCE COST { CPU_PER_SESSION CONNECT_TIME LOGICAL_READS_PER_SESSION PRIVATE_SGA } integer [{ CPU_PER_SESSION CONNECT_TIME LOGICAL_READS_PER_SESSION PRIVATE_SGA }] integer]... ;</pre>

SQL 문	구문
ALTER ROLE	<pre>ALTER ROLE role { NOT IDENTIFIED IDENTIFIED { BY password USING [schema.]package EXTERNALLY GLOBALLY } } ;</pre>
ALTER ROLEBACK SEGMENT	<pre>ALTER ROLLBACK SEGMENT rollback_segment { ONLINE OFFLINE storage_clause SHRINK [TO integer [K M]] } ;</pre>
ALTER SEQUENCE	<pre>ALTER SEQUENCE [schema.]sequence { INCREMENT BY integer { MAXVALUE integer NOMAXVALUE } { MINVALUE integer NOMINVALUE } { CYCLE NOCYCLE } { CACHE integer NOCACHE } { ORDER NOORDER } } [INCREMENT BY integer { MAXVALUE integer NOMAXVALUE } { MINVALUE integer NOMINVALUE } { CYCLE NOCYCLE } { CACHE integer NOCACHE } { ORDER NOORDER }]... ;</pre>
ALTER SESSION	<pre>ALTER SESSION { ADVISE { COMMIT ROLLBACK NOTHING } CLOSE DATABASE LINK dblink { ENABLE DISABLE } COMMIT IN PROCEDURE { ENABLE DISABLE } GUARD { ENABLE DISABLE FORCE } PARALLEL { DML DDL QUERY } [PARALLEL integer] { ENABLE RESUMABLE [TIMEOUT integer] [NAME string] DISABLE RESUMABLE } alter_session_set_clause } ;</pre>

SQL 문	구문
ALTER SYSTEM	<pre> ALTER SYSTEM { archive_log_clause checkpoint_clause check_datafiles_clause DUMP ACTIVE SESSION HISTORY [MINUTES integer] distributed_recov_clauses restricted_session_clauses FLUSH { SHARED_POOL BUFFER_CACHE } end_session_clauses SWITCH LOGFILE { SUSPEND RESUME } quiesce_clauses shutdown_dispatcher_clause REGISTER SET alter_system_set_clause [alter_system_set_clause]... RESET alter_system_reset_clause [alter_system_reset_clause]... } ; </pre>
ALTER TABLE	<pre> ALTER TABLE [schema.]table [alter_table_properties column_clauses constraint_clauses alter_table_partitioning alter_external_table_clauses move_table_clause] [enable_disable_clause { ENABLE DISABLE } { TABLE LOCK ALL TRIGGERS } [enable_disable_clause { ENABLE DISABLE } { TABLE LOCK ALL TRIGGERS }]...] ; </pre>
ALTER TABLESPACE	<pre> ALTER TABLESPACE tablespace { DEFAULT [table_compression] storage_clause MINIMUM EXTENT integer [K M] RESIZE size_clause COALESCE RENAME TO new_tablespace_name { BEGIN END } BACKUP datafile_tempfile_clauses tablespace_logging_clauses tablespace_group_clause tablespace_state_clause autoextend_clause flashback_mode_clause tablespace_retention_clause } ; </pre>

SQL 문	구문
ALTER TRIGGER	<pre>ALTER TRIGGER [schema.]trigger { ENABLE DISABLE RENAME TO new_name COMPILE [DEBUG] [compiler_parameters_clause [compiler_parameters_clause] ... [REUSE SETTINGS] } ;</pre>
ALTER TYPE	<pre>ALTER TYPE [schema.]type { compile_type_clause replace_type_clause { alter_method_spec alter_attribute_definition alter_collection_clauses [NOT] { INSTANTIABLE FINAL } } [dependent_handling_clause] } ;</pre>
ALTER USER	<pre>ALTER USER { user { IDENTIFIED { BY password [REPLACE old_password] EXTERNALLY GLOBALLY AS 'external_name' } DEFAULT TABLESPACE tablespace TEMPORARY TABLESPACE { tablespace tablespace_group_name } QUOTA { integer [K M] UNLIMITED } ON tablespace [QUOTA { integer [K M] UNLIMITED } ON tablespace]... PROFILE profile DEFAULT ROLE { role [, role]... ALL [EXCEPT role [, role]... NONE } PASSWORD EXPIRE ACCOUNT { LOCK UNLOCK } } [{ IDENTIFIED { BY password [REPLACE old_password] EXTERNALLY GLOBALLY AS 'external_name' } DEFAULT TABLESPACE tablespace TEMPORARY TABLESPACE { tablespace tablespace_group_name } QUOTA { integer [K M]</pre>

SQL 문	구문
	<pre> UNLIMITED } ON tablespace [QUOTA { integer [K M] UNLIMITED } ON tablespace]... PROFILE profile DEFAULT ROLE { role [, role]... ALL [EXCEPT role [, role]... NONE } PASSWORD EXPIRE ACCOUNT { LOCK UNLOCK } }]... user [, user]... proxy_clause ; </pre>
ALTER VIEW	<pre> ALTER VIEW [schema.]view { ADD out_of_line_constraint MODIFY CONSTRAINT constraint { RELY NORELY } DROP { CONSTRAINT constraint PRIMARY KEY UNIQUE (column [, column]...) } COMPILE } ; </pre>
ANALYZE	<pre> ANALYZE { TABLE [schema.]table [PARTITION (partition) SUBPARTITION (subpartition)] INDEX [schema.]index [PARTITION (partition) SUBPARTITION (subpartition)] CLUSTER [schema.]cluster } { validation_clauses LIST CHAINED ROWS [into_clause] DELETE [SYSTEM] STATISTICS compute_statistics_clause estimate_statistics_clause } ; </pre>
ASSOCIATE STATISTICS	<pre> ASSOCIATE STATISTICS WITH { column_association function_association } ; </pre>
AUDIT	<pre> AUDIT { sql_statement_clause schema_object_clause } [BY { SESSION ACCESS }] [WHENEVER [NOT] SUCCESSFUL] ; </pre>

SQL 문	구문
CALL	<pre>CALL { routine_clause object_access_expression } [INTO :host_variable [[INDICATOR] :indicator_variable]] ;</pre>
COMMENT	<pre>COMMENT ON { TABLE [schema.] { table view } COLUMN [schema.] { table. view. materialized_view. } column OPERATOR [schema.] operator INDEXTYPE [schema.] indextype MATERIALIZED VIEW materialized_view } IS 'text' ;</pre>
COMMIT	<pre>COMMIT [WORK] [COMMENT 'text' FORCE 'text' [, integer]] ;</pre>
CREATE CLUSTER	<pre>CREATE CLUSTER [schema.]cluster (column datatype [SORT] [, column datatype [SORT]]...) [{ physical_attributes_clause SIZE size_clause TABLESPACE tablespace { INDEX [SINGLE TABLE] HASHKEYS integer [HASH IS expr] } } [physical_attributes_clause SIZE size_clause TABLESPACE tablespace { INDEX [SINGLE TABLE] HASHKEYS integer [HASH IS expr] }] ...] [parallel_clause] [NOROWDEPENDENCIES ROWDEPENDENCIES] [CACHE NOCACHE] ;</pre>
CREATE CONTEXT	<pre>CREATE [OR REPLACE] CONTEXT namespace USING [schema.] package [INITIALIZED { EXTERNALLY GLOBALLY } ACCESSED GLOBALLY] ;</pre>

SQL 문	구문
CREATE CONTROLFILE	<pre> CREATE CONTROLFILE [REUSE] [SET] DATABASE database [logfile_clause] { RESETLOGS NORESETLOGS } [DATAFILE file_specification [, file_specification]...] [{ MAXLOGFILES integer MAXLOGMEMBERS integer MAXLOGHISTORY integer MAXDATAFILES integer MAXINSTANCES integer { ARCHIVELOG NOARCHIVELOG } FORCE LOGGING } [MAXLOGFILES integer MAXLOGMEMBERS integer MAXLOGHISTORY integer MAXDATAFILES integer MAXINSTANCES integer { ARCHIVELOG NOARCHIVELOG } FORCE LOGGING]...] [character_set_clause] ; </pre>
CREATE DATABASE	<pre> CREATE DATABASE [database] { USER SYS IDENTIFIED BY password USER SYSTEM IDENTIFIED BY password CONTROLFILE REUSE MAXDATAFILES integer MAXINSTANCES integer CHARACTER SET charset NATIONAL CHARACTER SET charset SET DEFAULT { BIGFILE SMALLFILE } TABLESPACE database_logging_clauses tablespace_clauses set_time_zone_clause }... ; </pre>
CREATE DATABASE LINK	<pre> CREATE [SHARED] [PUBLIC] DATABASE LINK dblink [CONNECT TO { CURRENT_USER user IDENTIFIED BY password [dblink_authentication] } dblink_authentication] [USING 'connect_string'] ; </pre>

SQL 문	구문
CREATE DIMENSION	<pre>CREATE DIMENSION [schema.]dimension level_clause [level_clause]... { hierarchy_clause attribute_clause extended_attribute_clause } [hierarchy_clause attribute_clause extended_attribute_clause]... ;</pre>
CREATE DIRECTORY	<pre>CREATE [OR REPLACE] DIRECTORY directory AS 'path_name' ;</pre>
CREATE DISKGROUP	<pre>CREATE DISKGROUP diskgroup_name [{ HIGH NORMAL EXTERNAL } REDUNDANCY] [FAILGROUP failgroup_name] DISK qualified_disk_clause [, qualified_disk_clause]... [[FAILGROUP failgroup_name] DISK qualified_disk_clause [, qualified_disk_clause]...]... ;</pre>
CREATE FUNCTION	<pre>CREATE [OR REPLACE] FUNCTION [schema.]function [(argument [IN OUT IN OUT] [NOCOPY] datatype [, argument [IN OUT IN OUT] [NOCOPY] datatype]...)] RETURN datatype [{ invoker_rights_clause DETERMINISTIC parallel_enable_clause } [invoker_rights_clause DETERMINISTIC parallel_enable_clause]...] { { AGGREGATE PIPELINED } USING [schema.]implementation_type [PIPELINED] { IS AS } { pl/sql_function_body call_spec } } ;</pre>

SQL 문	구문
CREATE INDEX	CREATE [UNIQUE BITMAP] INDEX [schema.]index ON { cluster_index_clause table_index_clause bitmap_join_index_clause } ;
CREATE INDEXTYPE	CREATE [OR REPLACE] INDEXTYPE [schema.]indextype FOR [schema.]operator (paramater_type [, paramater_type]...) [, [schema.]operator (paramater_type [, paramater_type]...)]... using_type_clause ;
CREATE JAVA	CREATE [OR REPLACE] [AND { RESOLVE COMPILE }] [NOFORCE] JAVA { { SOURCE RESOURCE } NAMED [schema.]primary_name CLASS [SCHEMA schema] } [invoker_rights_clause] [RESOLVER ((match_string [,] { schema_name - }) [(match_string [,] { schema_name - })]...)]...] { USING { BFILE (directory_object_name , server_file_name) { CLOB BLOB BFILE } subquery 'key_for_BLOB' } AS source_text } ;
CREATE LIBRARY	CREATE [OR REPLACE] LIBRARY [schema.]libname { IS AS } 'filename' [AGENT 'agent_dblink'] ;

Oracle Internal & Oracle Online Academy

SQL 문	구문
CREATE MATERIALIZED VIEW	<pre> CREATE MATERIALIZED VIEW [schema.]materialized_view [OF [schema.]object_type] [(scoped_table_ref_constraint)] { ON PREBUILT TABLE [{ WITH WITHOUT } REDUCED PRECISION] physical_properties materialized_view_props } [USING INDEX [physical_attributes_clause TABLESPACE tablespace] [physical_attributes_clause TABLESPACE tablespace]... USING NO INDEX] [create_mv_refresh] [FOR UPDATE] [{ DISABLE ENABLE } QUERY REWRITE] AS subquery ; </pre>
CREATE MATERIALIZED VIEW LOG	<pre> CREATE MATERIALIZED VIEW LOG ON [schema.] table [physical_attributes_clause TABLESPACE tablespace logging_clause { CACHE NOCACHE } [physical_attributes_clause TABLESPACE tablespace logging_clause { CACHE NOCACHE }]...] [parallel_clause] [table_partitioning_clauses] [WITH { OBJECT ID PRIMARY KEY ROWID SEQUENCE (column [, column]...) } [, { OBJECT ID PRIMARY KEY ROWID SEQUENCE (column [, column]...) }]... [new_values_clause] ; </pre>

SQL 문	구문
CREATE OPERATOR	CREATE [OR REPLACE] OPERATOR [schema.] operator binding_clause ;
CREATE OUTLINE	CREATE [OR REPLACE] [PUBLIC PRIVATE] OUTLINE [outline] [FROM [PUBLIC PRIVATE] source_outline] [FOR CATEGORY category] [ON statement] ;
CREATE PACKAGE	CREATE [OR REPLACE] PACKAGE [schema.]package [invoker_rights_clause] { IS AS } pl/sql_package_spec ;
CREATE PACKAGE BODY	CREATE [OR REPLACE] PACKAGE BODY [schema.]package { IS AS } pl/sql_package_body ;
CREATE PFILE	CREATE PFILE [= 'pfile_name'] FROM SPFILE [= 'spfile_name'] ;
CREATE PROCEDURE	CREATE [OR REPLACE] PROCEDURE [schema.]procedure [(argument [IN OUT IN OUT] [NOCOPY] datatype [, argument [IN OUT IN OUT] [NOCOPY] datatype]...)] [invoker_rights_clause] { IS AS } { pl/sql_subprogram_body call_spec } ;
CREATE PROFILE	CREATE PROFILE profile LIMIT { resource_parameters password_parameters } [resource_parameters password_parameters]... ;
CREATE ROLE	CREATE ROLE role [NOT IDENTIFIED IDENTIFIED { BY password USING [schema.] package EXTERNALLY GLOBALLY }]

SQL 문	구문
CREATE ROLEBACK SEGMENT	CREATE [PUBLIC] ROLLBACK SEGMENT rollback_segment [{ TABLESPACE tablespace storage_clause } TABLESPACE tablespace storage_clause]...];
CREATE SCHEMA	CREATE SCHEMA AUTHORIZATION schema { create_table_statement create_view_statement grant_statement } [create_table_statement create_view_statement grant_statement]... ;
CREATE SEQUENCE	CREATE SEQUENCE [schema.]sequence [{ INCREMENT BY START WITH } integer { MAXVALUE integer NOMAXVALUE } { MINVALUE integer NOMINVALUE } { CYCLE NOCYCLE } { CACHE integer NOCACHE } { ORDER NOORDER }] [{ INCREMENT BY START WITH } integer { MAXVALUE integer NOMAXVALUE } { MINVALUE integer NOMINVALUE } { CYCLE NOCYCLE } { CACHE integer NOCACHE } { ORDER NOORDER }]... ;
CREATE SPFILE	CREATE SPFILE [= 'spfile_name'] FROM PFILE [= 'pfile_name'] ;
CREATE SYNONYM	CREATE [OR REPLACE] [PUBLIC] SYNONYM [schema.]synonym FOR [schema.]object [@ dblink] ;
CREATE TABLE	{ relational_table object_table XMLType_table }
CREATE TABLESPACE	CREATE [BIGFILE SMALLFILE] { permanent_tablespace_clause temporary_tablespace_clause undo_tablespace_clause } ;

SQL 문	구문
CREATE TRIGGER	<pre> CREATE [OR REPLACE] TRIGGER [schema.]trigger { BEFORE AFTER INSTEAD OF } { dml_event_clause { ddl_event [OR ddl_event]... database_event [OR database_event]... } ON { [schema.]SCHEMA DATABASE } } [WHEN (condition)] { pl/sql_block call_procedure_statement } ; </pre>
CREATE TYPE	<pre> { create_incomplete_type create_object_type create_varray_type create_nested_table_type } </pre>
CREATE TYPE BODY	<pre> CREATE [OR REPLACE] TYPE BODY [schema.]type_name { IS AS } { subprogram_declaration map_order_func_declaration } [; { subprogram_declaration map_order_func_declaration }] [...] END ; </pre>
CREATE USER	<pre> CREATE USER user IDENTIFIED { BY password EXTERNALLY GLOBALLY AS 'external_name' } [DEFAULT TABLESPACE tablespace TEMPORARY TABLESPACE { tablespace tablespace_group_name } QUOTA { integer [K M] UNLIMITED } ON tablespace [QUOTA { integer [K M] UNLIMITED } ON tablespace] PROFILE profile PASSWORD EXPIRE ACCOUNT { LOCK UNLOCK } [DEFAULT TABLESPACE tablespace TEMPORARY TABLESPACE { tablespace tablespace_group_name } QUOTA { integer [K M] UNLIMITED }] </pre>

SQL 문	구문
	<pre> ON tablespace [QUOTA { integer [K M] UNLIMITED } ON tablespace]... PROFILE profile PASSWORD EXPIRE ACCOUNT { LOCK UNLOCK }]...] ; </pre>
CREATE VIEW	<pre> CREATE [OR REPLACE] [[NO] FORCE] VIEW [schema.]view [(alias [inline_constraint [inline_constraint]...] out_of_line_constraint [, alias [inline_constraint [inline_constraint]...] out_of_line_constraint]...) object_view_clause XMLType_view_clause] AS subquery [subquery_restriction_clause] ; </pre>
DELETE	<pre> DELETE [hint] [FROM] { dml_table_expression_clause ONLY (dml_table_expression_clause) } [t_alias] [where_clause] [returning_clause] ; </pre>
DISASSOCIATE STATISTICS	<pre> DISASSOCIATE STATISTICS FROM { COLUMNS [schema.]table.column [, [schema.]table.column]... FUNCTIONS [schema.]function [, [schema.]function]... PACKAGES [schema.]package [, [schema.]package]... TYPES [schema.]type [, [schema.]type]... INDEXES [schema.]index [, [schema.]index]... INDEXTYPES [schema.]indextype [, [schema.]indextype]... } [FORCE] ; </pre>
DROP CLUSTER	<pre> DROP CLUSTER [schema.]cluster [INCLUDING TABLES [CASCADE CONSTRAINTS]] ; </pre>
DROP CONTEXT	<pre> DROP CONTEXT namespace ; </pre>

SQL 문	구문
DROP DATABASE	DROP DATABASE ;
DROP DATABASE LINK	DROP [PUBLIC] DATABASE LINK dblink ;
DROP DIMENSION	DROP DIMENSION [schema.]dimension ;
DROP DIRECTORY	DROP DIRECTORY directory_name ;
DROP DISKGROUP	DROP DISKGROUP diskgroup_name [{ INCLUDING EXCLUDING } CONTENTS] ;
DROP FUNCTION	DROP FUNCTION [schema.]function_name ;
DROP INDEX	DROP INDEX [schema.]index [FORCE] ;
DROP INDEXTYPE	DROP INDEXTYPE [schema.]inextype [FORCE] ;
DROP JAVA	DROP JAVA { SOURCE CLASS RESOURCE } [schema.]object_name ;
DROP LIBRARY	DROP LIBRARY library_name ;
DROP MATERIALIZED VIEW	DROP MATERIALIZED VIEW [schema.]materialized_view [PRESERVE TABLE] ;
DROP MATERIALIZED VIEW LOG	DROP MATERIALIZED VIEW LOG ON [schema.]table ;
DROP OPERATOR	DROP OPERATOR [schema.]operator [FORCE] ;
DROP OUTLINE	DROP OUTLINE outline ;
DROP PACKAGE	DROP PACKAGE [BODY] [schema.]package ;
DROP PROCEDURE	DROP PROCEDURE [schema.]procedure ;
DROP PROFILE	DROP PROFILE profile [CASCADE] ;
DROP ROLE	DROP ROLE role ;
DROP ROLLBACK SEGMENT	DROP ROLLBACK SEGMENT rollback_segment ;
DROP SEQUENCE	DROP SEQUENCE [schema.]sequence_name ;

SQL 문	구문
DROP SYNONYM	DROP [PUBLIC] SYNONYM [schema.]synonym [FORCE] ;
DROP TABLE	DROP TABLE [schema.]table [CASCADE CONSTRAINTS] [PURGE] ;
DROP TABLESPACE	DROP TABLESPACE tablespace [INCLUDING CONTENTS [AND DATAFILES] [CASCADE CONSTRAINTS]] ;
DROP TRIGGER	DROP TRIGGER [schema.]trigger ;
DROP TYPE	DROP TYPE [schema.]type_name [FORCE VALIDATE] ;
DROP TYPE BODY	DROP TYPE BODY [schema.]type_name ;
DROP USER	DROP USER user [CASCADE] ;
DROP VIEW	DROP VIEW [schema.] view [CASCADE CONSTRAINTS] ;
EXPLAIN PLAN	EXPLAIN PLAN [SET STATEMENT_ID = 'text'] [INTO [schema.]table [@ dblink]] FOR statement ;
FLASHBACK DATABASE	FLASHBACK [STANDBY] DATABASE [database] { TO { SCN TIMESTAMP } expr TO BEFORE { SCN TIMESTAMP } expr };
FLASHBACK TABLE	FLASHBACK TABLE [schema.]table [, [schema.]table]... TO { { SCN TIMESTAMP } expr [{ ENABLE DISABLE } TRIGGERS] BEFORE DROP [RENAME TO table] } ;
GRANT	GRANT { grant_system_privileges grant_object_privileges } ;
INSERT	INSERT [hint] { single_table_insert multi_table_insert } ;

SQL 문	구문
LOCK TABLE	LOCK TABLE [schema.] { table view } [{ PARTITION (partition) SUBPARTITION (subpartition) } @ dblink] [, [schema.] { table view } [{ PARTITION (partition) SUBPARTITION (subpartition) } @ dblink]]]... IN lockmode MODE [NOWAIT] ;
MERGE	MERGE [hint] INTO [schema.]table [t_alias] USING [schema.] { table view subquery } [t_alias] ON (condition) [merge_update_clause] [merge_insert_clause] ;
NOAUDIT	NOAUDIT { sql_statement_clause [, sql_statement_clause]... schema_object_clause [, schema_object_clause]... } [WHENEVER [NOT] SUCCESSFUL] ;
PURGE	PURGE { { TABLE table INDEX index } { RECYCLEBIN DBA_RECYCLEBIN } TABLESPACE tablespace [USER user] } ;
RENAME	RENAME old_name TO new_name ;
REVOKE	REVOKE { revoke_system_privileges revoke_object_privileges } ;
ROLLBACK	ROLLBACK [WORK] [TO [SAVEPOINT] savepoint FORCE 'text'] ;
SAVEPOINT	SAVEPOINT savepoint ;

SQL 문	구문
SELECT	subquery [for_update_clause] ;
SET CONSTRAINT[S]	SET { CONSTRAINT CONSTRAINTS } { constraint [, constraint]... ALL } { IMMEDIATE DEFERRED } ;
SET ROLE	SET ROLE { role [IDENTIFIED BY password] [, role [IDENTIFIED BY password]]... ALL [EXCEPT role [, role]...] NONE } ;
SET TRANSACTION	SET TRANSACTION { { READ { ONLY WRITE } ISOLATION LEVEL { SERIALIZABLE READ COMMITTED } USE ROLLBACK SEGMENT rollback_segment } [NAME 'text'] NAME 'text' } ;
TRUNCATE	TRUNCATE { TABLE [schema.]table [{ PRESERVE PURGE } MATERIALIZED VIEW LOG] CLUSTER [schema.]cluster } [{ DROP REUSE } STORAGE] ;
UPDATE	UPDATE [hint] { dml_table_expression_clause ONLY (dml_table_expression_clause) } [t_alias] update_set_clause [where_clause] [returning_clause] ;

Oracle Internal & Oracle Online Academy

표 2: 하위 절 구문

하위 절	구문
activate_standby_db_clause	ACTIVATE [PHYSICAL LOGICAL] STANDBY DATABASE [SKIP [STANDBY LOGFILE]]
add_binding_clause	ADD BINDING (parameter_type [, parameter_type]...) RETURN (return_type) [implementation_clause] using_function_clause
add_column_clause	ADD (column datatype [DEFAULT expr] [{ inline_constraint [inline_constraint]... inline_ref_constraint }] , column datatype [DEFAULT expr] [{ inline_constraint [inline_constraint]... inline_ref_constraint }]]] [column_properties]
add_disk_clause	ADD [FAILGROUP failgroup_name] DISK qualified_disk_clause [, qualified_disk_clause]... [[FAILGROUP failgroup_name] DISK qualified_disk_clause [, qualified_disk_clause]...]]
add_hash_index_partition	ADD PARTITION [partition_name] [TABLESPACE tablespace_name] [parallel_clause]
add_hash_partition_clause	ADD PARTITION [partition] partitioning_storage_clause [update_index_clauses] [parallel_clause]

하위 절	구문
add_hash_subpartition	ADD subpartition_spec [update_index_clauses] [parallel_clause]
add_list_partition_clause	ADD PARTITION [partition] list_values_clause [table_partition_description] [update_index_clauses]
add_list_subpartition	ADD subpartition_spec [update_index_clauses]
add_logfile_clauses	ADD [STANDBY] LOGFILE { [INSTANCE 'instance_name' THREAD integer] [GROUP integer] redo_log_file_spec [, [GROUP integer] redo_log_file_spec]... MEMBER 'filename' [REUSE] [, 'filename' [REUSE]]... TO logfile_descriptor [, logfile_descriptor]... }
add_overflow_clause	ADD OVERFLOW [segment_attributes_clause] [(PARTITION [segment_attributes_clause] [, PARTITION [segment_attributes_clause]]...)]
add_range_partition_clause	ADD PARTITION [partition] range_values_clause [table_partition_description] [update_index_clauses]
add_table_partition	{ add_range_partition_clause add_hash_partition_clause add_list_partition_clause }
alias_file_name	+diskgroup_name [(template_name)] /alias_name
allocate_extent_clause	ALLOCATE EXTENT [({ SIZE size_clause DATAFILE 'filename' INSTANCE integer } [SIZE size_clause DATAFILE 'filename' INSTANCE integer]...)]

하위 절	구문
alter_attribute_definition	{ { ADD MODIFY } ATTRIBUTE { attribute [datatype] (attribute datatype [, attribute datatype]...) } DROP ATTRIBUTE { attribute (attribute [, attribute]...) } }
alter_collection_clauses	MODIFY { LIMIT integer ELEMENT TYPE datatype }
alter_datafile_clause	DATAFILE { 'filename' filenumber } [, 'filename' filenumber]... } { ONLINE OFFLINE [FOR DROP] RESIZE size_clause autoextend_clause END BACKUP }
alter_external_table_clauses	{ add_column_clause modify_column_clauses drop_column_clause parallel_clause external_data_properties REJECT LIMIT { integer UNLIMITED } PROJECT COLUMN { ALL REFERENCED } } [add_column_clause modify_column_clauses drop_column_clause parallel_clause external_data_properties REJECT LIMIT { integer UNLIMITED } PROJECT COLUMN { ALL REFERENCED }]...
alter_index_partitioning	{ modify_index_defaultAttrs add_hash_index_partition modify_index_partition rename_index_partition drop_index_partition split_index_partition coalesce_index_partition modify_index_subpartition }

하위 절	구문
alter_iot_clauses	{ index_org_table_clause alter_overflow_clause alter_mapping_table_clauses COALESCE }
alter_mapping_table_clauses	MAPPING TABLE { UPDATE BLOCK REFERENCES allocate_extent_clause deallocate_unused_clause }
alter_method_spec	{ ADD DROP } { map_order_function_spec subprogram_spec } [{ ADD DROP } { map_order_function_spec subprogram_spec }]...]
alter_mv_refresh	REFRESH { { FAST COMPLETE FORCE } ON { DEMAND COMMIT } { START WITH NEXT } date WITH PRIMARY KEY USING { DEFAULT MASTER ROLLBACK SEGMENT MASTER ROLLBACK SEGMENT rollback_segment } USING { ENFORCED TRUSTED } CONSTRAINTS }
alter_overflow_clause	{ OVERFLOW { allocate_extent_clause deallocate_unused_clause } [allocate_extent_clause deallocate_unused_clause]]... add_overflow_clause }
alter_session_set_clause	SET parameter_name = parameter_value [parameter_name = parameter_value]...
alter_system_reset_clause	parameter_name [SCOPE = { MEMORY SPFILE BOTH }] SID = 'sid'

하위 절	구문
alter_system_set_clause	<pre> parameter_name = parameter_value [, parameter_value]... [COMMENT 'text'] [DEFERRED] [SCOPE = { MEMORY SPFILE BOTH }] [SID = { 'sid' * }] </pre>
alter_table_partitioning	<pre> { modify_table_defaultAttrs set_subpartition_template modify_table_partition modify_table_subpartition move_table_partition move_table_subpartition add_table_partition coalesce_table_partition drop_table_partition drop_table_subpartition rename_partition_subpart truncate_partition_subpart split_table_partition split_table_subpartition merge_table_partitions merge_table_subpartitions exchange_partition_subpart } </pre>
alter_table_properties	<pre> { physical_attributes_clause logging_clause table_compression supplemental_table_logging allocate_extent_clause deallocate_unused_clause shrink_clause { CACHE NOCACHE } upgrade_table_clause records_per_block_clause parallel_clause row_movement_clause } [physical_attributes_clause logging_clause table_compression supplemental_table_logging allocate_extent_clause deallocate_unused_clause shrink_clause { CACHE NOCACHE } upgrade_table_clause records_per_block_clause parallel_clause row_movement_clause]... RENAME TO new_table_name } [alter_iot_clauses] </pre>

하위 절	구문
alter_tempfile_clause	<pre> TEMPFILE { 'filename' [, 'filename']... filenumber [, filenumber]... } { RESIZE size_clause autoextend_clause DROP [INCLUDING DATAFILES] ONLINE OFFLINE } </pre>
alter_varray_col_properties	<pre> MODIFY VARRAY varray_item (modify_LOB_parameters) </pre>
analytic_clause	<pre> [query_partition_clause] [order_by_clause [windowing_clause]] </pre>
archive_log_clause	<pre> ARCHIVE LOG [INSTANCE 'instance_name' THREAD integer] { { SEQUENCE integer CHANGE integer CURRENT [NOSWITCH] GROUP integer LOGFILE 'filename' [USING BACKUP CONTROLFILE] NEXT ALL START } [TO 'location'] STOP } </pre>
array_DML_clause	<pre> [WITH WITHOUT] ARRAY DML [([schema.]type [, [schema.]varray_type]) [, ([schema.]type [, [schema.]varray_type])...] </pre>
attribute_clause	<pre> ATTRIBUTE level DETERMINES { dependent_column (dependent_column [, dependent_column]...) } </pre>
auditing_by_clause	<pre> BY { proxy [, proxy]... user [, user]... } </pre>

하위 절	구문
auditing_on_clause	ON { [schema.]object DIRECTORY directory_name DEFAULT }
autoextend_clause	AUTOEXTEND { OFF ON [NEXT size_clause] [maxsize_clause] }
binding_clause	BINDING (parameter_type [, parameter_type]...) RETURN return_type [implementation_clause] using_function_clause , (parameter_type [, parameter_type]...) RETURN return_type [implementation_clause] using_function_clause]...]
bitmap_join_index_clause	[schema.]table ([[schema.]table. t_alias.]column [ASC DESC] [, [[schema.]table. t_alias.]column [ASC DESC]]...) FROM [schema.]table [t_alias] [, [schema.]table [t_alias]]... WHERE condition [local_partitioned_index] index_attributes
build_clause	BUILD { IMMEDIATE DEFERRED }
C_declaration	C [NAME name] LIBRARY lib_name [AGENT IN (argument[, argument]...)] [WITH CONTEXT] [PARAMETERS (parameter[, parameter]...)]
call_spec	LANGUAGE { Java_declaration C_declaration }
cancel_clause	CANCEL [IMMEDIATE] [WAIT NOWAIT]

하위 절	구문
cell_assignment	<pre> measure_column [{ { condition expr single_column_for_loop } [, { condition expr single_column_for_loop }] ... multi_column_for_loop }] </pre> <p>Note: The outer square brackets are part of the syntax. In this case, they do not indicate optionality.</p>
cell_reference_options	[{ IGNORE KEEP } NAV] [UNIQUE { DIMENSION SINGLE REFERENCE }]
character_set_clause	CHARACTER SET character_set
check_datafiles_clause	CHECK DATAFILES [GLOBAL LOCAL]
check_diskgroup_clauses	<pre> CHECK { ALL DISK disk_name [, disk_name]... DISKS IN FAILGROUP failgroup_name [, failgroup_name]... FILE filename [, filename]... } [CHECK { ALL DISK disk_name [, disk_name]... DISKS IN FAILGROUP failgroup_name [, failgroup_name]... FILE filename [, filename]... }] ... [REPAIR NOREPAIR] </pre>
checkpoint_clause	CHECKPOINT [GLOBAL LOCAL]
cluster_index_clause	CLUSTER [schema.] cluster index_attributes

하위 절	구문
coalesce_index_partition	COALESCE PARTITION [parallel_clause]
coalesce_table_partition	COALESCE PARTITION [update_index_clauses] [parallel_clause]
column_association	COLUMNS [schema.]table.column [, [schema.]table.column]... using_statistics_type
column_clauses	{ { add_column_clause modify_column_clause drop_column_clause } [add_column_clause modify_column_clause drop_column_clause]... rename_column_clause modify_collection_retrieval [modify_collection_retrieval]... modify_LOB_storage_clause alter_varray_col_properties }
column_properties	{ object_type_col_properties nested_table_col_properties { varray_col_properties LOB_storage_clause } (LOB_partition_storage [, LOB_partition_storage]...) XMLType_column_properties } [{ object_type_col_properties nested_table_col_properties { varray_col_properties LOB_storage_clause } (LOB_partition_storage [, LOB_partition_storage]...)] XMLType_column_properties }]...
commit_switchover_clause	{ PREPARE COMMIT } TO SWITCHOVER [TO { { PHYSICAL LOGICAL } PRIMARY PHYSICAL STANDBY [{ WITH WITHOUT } SESSION SHUTDOWN { WAIT NOWAIT }] LOGICAL STANDBY } CANCEL]

하위 절	구문
compile_type_clause	COMPILE [DEBUG] [SPECIFICATION BODY] [compiler_parameters_clause [compiler_parameters_clause] ...] [REUSE SETTINGS]
compiler_parameters_clause	parameter_name = parameter_value
composite_partitioning	PARTITION BY RANGE (column_list) [subpartition_by_list subpartition_by_hash] (PARTITION [partition] range_values_clause table_partition_description [, PARTITION [partition] range_values_clause table_partition_description] ...)
compute_statistics_clause	COMPUTE [SYSTEM] STATISTICS [for_clause]
conditional_insert_clause	[ALL FIRST] WHEN condition THEN insert_into_clause [values_clause] [error_logging_clause] [insert_into_clause [values_clause] [error_logging_clause]]... [WHEN condition THEN insert_into_clause [values_clause] [error_logging_clause] [insert_into_clause [values_clause] [error_logging_clause]]...]... [ELSE insert_into_clause [values_clause] [error_logging_clause] [insert_into_clause [values_clause] [error_logging_clause]]...]
constraint	{ inline_constraint out_of_line_constraint inline_ref_constraint out_of_line_ref_constraint }

하위 절	구문
constraint_clauses	{ ADD { out_of_line_constraint [out_of_line_constraint]... out_of_line_REF_constraint } MODIFY { CONSTRAINT constraint PRIMARY KEY UNIQUE (column [, column]...) } constraint_state RENAME CONSTRAINT old_name TO new_name drop_constraint_clause }
constraint_state	[[NOT] DEFERRABLE] [INITIALLY { IMMEDIATE DEFERRED }] [INITIALLY { IMMEDIATE DEFERRED }] [[NOT] DEFERRABLE]] [RELY NORELY] [using_index_clause] [ENABLE DISABLE] [VALIDATE NOVALIDATE] [exceptions_clause]
constructor_declaration	[FINAL] [INSTANTIABLE] CONSTRUCTOR FUNCTION datatype [[SELF IN OUT datatype,] parameter datatype [, parameter datatype]...] RETURN SELF AS RESULT { IS AS } { pl/sql_block call_spec }
constructor_spec	[FINAL] [INSTANTIABLE] CONSTRUCTOR FUNCTION datatype [([SELF IN OUT datatype,] parameter datatype [, parameter datatype]...)] RETURN SELF AS RESULT [{ IS AS } call_spec]
context_clause	[WITH INDEX CONTEXT, SCAN CONTEXT implementation_type [COMPUTE ANCILLARY DATA]] [WITH COLUMN CONTEXT]

하위 절	구문
controlfile_clauses	{ CREATE [LOGICAL PHYSICAL] STANDBY CONTROLFILE AS 'filename' [REUSE] BACKUP CONTROLFILE TO { 'filename' [REUSE] trace_file_clause } }
create_datafile_clause	CREATE DATAFILE { 'filename' filenumber } [, 'filename' filenumber]... } [AS { file_specification [, file_specification]... NEW }]
create_incomplete_type	CREATE [OR REPLACE] TYPE [schema.]type_name ;
create_mv_refresh	{ REFRESH { { FAST COMPLETE FORCE } ON { DEMAND COMMIT } { START WITH NEXT } date WITH { PRIMARY KEY ROWID } USING { DEFAULT [MASTER LOCAL] ROLLBACK SEGMENT [MASTER LOCAL] ROLLBACK SEGMENT rollback_segment } [DEFAULT [MASTER LOCAL] ROLLBACK SEGMENT [MASTER LOCAL] ROLLBACK SEGMENT rollback_segment]... USING { ENFORCED TRUSTED } CONSTRAINTS } { { FAST COMPLETE FORCE } ON { DEMAND COMMIT } { START WITH NEXT } date WITH { PRIMARY KEY ROWID } USING { DEFAULT [MASTER LOCAL] ROLLBACK SEGMENT [MASTER LOCAL] ROLLBACK SEGMENT rollback_segment } [DEFAULT [MASTER LOCAL] ROLLBACK SEGMENT [MASTER LOCAL] ROLLBACK SEGMENT rollback_segment]... }

하위 절	구문
	<pre> USING { ENFORCED TRUSTED } CONSTRAINTS]... NEVER REFRESH } </pre>
create_nested_table_type	<pre> CREATE [OR REPLACE] TYPE [schema.]type_name [OID 'object_identifier'] { IS AS } TABLE OF datatype ; </pre>
create_object_type	<pre> CREATE [OR REPLACE] TYPE [schema.]type_name [OID 'object_identifier'] [invoker_rights_clause] { { IS AS } OBJECT UNDER [schema.]supertype } [sqlj_object_type] [(attribute datatype [sqlj_object_type_attr] [, attribute datatype [sqlj_object_type_attr]... [, element_spec [, element_spec]...])] [[NOT] FINAL] [[NOT] INSTANTIABLE] ; </pre>
create_varray_type	<pre> CREATE [OR REPLACE] TYPE [schema.]type_name [OID 'object_identifier'] { IS AS } { VARRAY VARYING ARRAY } (limit) OF datatype ; </pre>
database_file_clauses	<pre> { RENAME FILE 'filename' [, 'filename']... TO 'filename' create_datafile_clause alter_datafile_clause alter_tempfile_clause } </pre>
database_logging_clauses	<pre> { LOGFILE [GROUP integer] file_specification [, [GROUP integer] file_specification]... MAXLOGFILES integer MAXLOGMEMBERS integer MAXLOGHISTORY integer { ARCHIVELOG NOARCHIVELOG } FORCE LOGGING } </pre>

하위 절	구문
datafile_tempfile_clauses	{ ADD { DATAFILE TEMPFILE } [file_specification [, file_specification]...] RENAME DATAFILE 'filename' [, 'filename']... TO 'filename' [, 'filename']... { DATAFILE TEMPFILE } { ONLINE OFFLINE } }
datafile_tempfile_spec	['filename' [SIZE size_clause] [REUSE] [autoextend_clause]
dblink	database[.domain [.domain]...] [@ connect_descriptor]
dblink_authentication	AUTHENTICATED BY user IDENTIFIED BY password
deallocate_unused_clause	DEALLOCATE UNUSED [KEEP size_clause]
default_cost_clause	DEFAULT COST (cpu_cost, io_cost, network_cost)
default_selectivity_clause	DEFAULT SELECTIVITY default_selectivity
default_tablespace	DEFAULT TABLESPACE tablespace [DATAFILE datafile_tempfile_spec] extent_management_clause
default_settings_clauses	{ SET DEFAULT { BIGFILE SMALLFILE } TABLESPACE DEFAULT TABLESPACE tablespace DEFAULT TEMPORARY TABLESPACE { tablespace tablespace_group_name } RENAME GLOBAL_NAME TO database.domain [.domain]... { ENABLE BLOCK CHANGE TRACKING [USING FILE 'filename' [REUSE]] DISABLE BLOCK CHANGE TRACKING } flashback_mode_clause set_time_zone_clause }
default_temp_tablespace	[BIGFILE SMALLFILE] DEFAULT TEMPORARY TABLESPACE tablespace [TEMPFILE file_specification [, file_specification]...] extent_management_clause

하위 절	구문
dependent_handling_clause	{ INVALIDATE CASCADE [{ [NOT] INCLUDING TABLE DATA CONVERT TO SUBSTITUTABLE }] [[FORCE] exceptions_clause] }
dimension_join_clause	JOIN KEY { child_key_column (child_key_column [, child_key_column]...) } REFERENCES parent_level [JOIN KEY { child_key_column (child_key_column [, child_key_column]...) } REFERENCES parent_level]...
disk_clauses	{ diskgroup_name { add_disk_clause drop_disk_clauses resize_disk_clauses } { diskgroup_name ALL } undrop_disk_clause }
diskgroup_alias_clauses	{ ADD ALIAS alias_name FOR filename [, alias_name FOR filename]... DROP ALIAS alias_name [, alias_name]... RENAME ALIAS old_alias_name TO new_alias_name [, old_alias_name TO new_alias_name]...
diskgroup_availability	{ MOUNT DISMOUNT [FORCE NOFORCE] }
diskgroup_clauses	{ diskgroup_name { rebalance_diskgroup_clause check_diskgroup_clauses diskgroup_template_clauses diskgroup_directory_clauses diskgroup_alias_clauses drop_diskgroup_file_clause } { diskgroup_name ALL } diskgroup_availability }

하위 절	구문
diskgroup_directory_clauses	{ ADD DIRECTORY filename [, filename]... DROP DIRECTORY filename [FORCE NOFORCE] [, filename [FORCE NOFORCE]]... RENAME DIRECTORY old_dir_name TO new_dir_name [, old_dir_name TO new_dir_name]... }
diskgroup_file_spec	[' { fully_qualified_file_name numeric_file_name incorporate_file_name alias_file_name } , [SIZE size_clause] [REUSE] [autoextend_clause]
diskgroup_template_clauses	{ { ADD ALTER } TEMPLATE qualified_template_clause [, qualified_template_clause]... DROP TEMPLATE template_name [, template_name]... }
distributed_recov_clauses	{ ENABLE DISABLE } DISTRIBUTED RECOVERY
dml_event_clause	{ DELETE INSERT UPDATE [OF column [, column]...] } [OR { DELETE INSERT UPDATE [OF column [, column]...] }] [...] ON { [schema.] table [NESTED TABLE nested_table_column OF] [schema.] view } [referencing_clause] [FOR EACH ROW]
dml_table_expression_clause	{ [schema.] { table [{ PARTITION (partition) SUBPARTITION (subpartition) } @ dblink] { view materialized view } [@ dblink] } (subquery [subquery_restriction_clause]) table_collection_expression }

하위 절	구문
domain_index_clause	INDEXTYPE IS indextype [parallel_clause] [PARAMETERS ('ODCI_parameters')]
drop_binding_clause	DROP BINDING (parameter_type [, parameter_type]...) [FORCE]
drop_column_clause	{ SET UNUSED { COLUMN column (column [, column]...) } { CASCADE CONSTRAINTS INVALIDATE } { CASCADE CONSTRAINTS INVALIDATE }... } DROP { COLUMN column (column [, column]...) } { CASCADE CONSTRAINTS INVALIDATE } { CASCADE CONSTRAINTS INVALIDATE }... [CHECKPOINT integer] DROP { UNUSED COLUMNS COLUMNS CONTINUE } [CHECKPOINT integer] }
drop_constraint_clause	DROP { { PRIMARY KEY UNIQUE (column [, column]...) } [CASCADE] { KEEP DROP } INDEX } CONSTRAINT constraint [CASCADE] }
drop_disk_clauses	DROP { DISK disk_name [FORCE NOFORCE] [, disk_name [FORCE NOFORCE]]... DISKS IN FAILGROUP failgroup_name [FORCE NOFORCE] [, failgroup_name [FORCE NOFORCE]]... }
drop_diskgroup_file_clause	DROP FILE filename [, filename]...
drop_index_partition	DROP PARTITION partition_name

하위 절	구문
drop_logfile_clauses	DROP [STANDBY] LOGFILE { logfile_descriptor [, logfile_descriptor]... MEMBER 'filename' [, 'filename']... }
drop_table_partition	DROP PARTITION partition [update_index_clauses [parallel_clause]]
drop_table_subpartition	DROP SUBPARTITION subpartition [update_index_clauses [parallel_clause]]
element_spec	[inheritance_clauses] { subprogram_spec constructor_spec map_order_function_spec } [subprogram_clause constructor_spec map_order_function_spec]... [, pragma_clause]
else_clause	ELSE else_expr
enable_disable_clause	{ ENABLE DISABLE } [VALIDATE NOVALIDATE] { UNIQUE (column [, column]...) PRIMARY KEY CONSTRAINT constraint } [using_index_clause] [exceptions_clause] [CASCADE] [{ KEEP DROP } INDEX]
end_session_clauses	{ DISCONNECT SESSION 'integer1, integer2' [POST_TRANSACTION] KILL SESSION 'integer1, integer2' } [IMMEDIATE]
estimate_statistics_clause	ESTIMATE [SYSTEM] STATISTICS [for_clause] [SAMPLE integer { ROWS PERCENT }]
exceptions_clause	EXCEPTIONS INTO [schema.]table
exchange_partition_subpart	EXCHANGE { PARTITION partition SUBPARTITION subpartition } WITH TABLE table [{ INCLUDING EXCLUDING } INDEXES] [{ WITH WITHOUT } VALIDATION] [exceptions_clause] [update_index_clauses [parallel_clause]]

하위 절	구문
expr	{ simple_expression compound_expression case_expression cursor_expression datetime_expression function_expression interval_expression object_access_expression scalar_subquery_expression model_expression type_constructor_expression variable_expression }
expression_list	{ expr [, expr]... (expr [, expr]...) }
extended_attribute_clause	ATTRIBUTE attribute LEVEL level DETERMINES { dependent_column (dependent_column [, dependent_column]...) [LEVEL level DETERMINES { dependent_column (dependent_column [, dependent_column]...)]]... }
extent_management_clause	EXTENT MANAGEMENT { DICTIONARY LOCAL [AUTOALLOCATE UNIFORM [SIZE size_clause]] }
external_data_properties	DEFAULT DIRECTORY directory [ACCESS PARAMETERS { (opaque_format_spec) USING CLOB subquery }] LOCATION ([directory:] 'locationSpecifier' [, [directory:] 'locationSpecifier']...)

하위 절	구문
external_table_clause	([TYPE access_driver_type] external_data_properties) [REJECT LIMIT { integer UNLIMITED }]
file_specification	{ datafile_tempfile_spec diskgroup_file_spec redo_log_file_spec }
finish_clause	[DISCONNECT [FROM SESSION]] [parallel_clause] FINISH [SKIP [STANDBY LOGFILE]] [WAIT NOWAIT]
flashback_mode_clause	FLASHBACK { ON OFF }
flashback_query_clause	[VERSIONS BETWEEN { SCN TIMESTAMP } { expr MINVALUE } AND { expr MAXVALUE }] AS OF { SCN TIMESTAMP } expr
for_clause	FOR { TABLE ALL [INDEXED] COLUMNS [SIZE integer] COLUMNS [SIZE integer] { column attribute } [SIZE integer] [{ column attribute } [SIZE integer]]... ALL [LOCAL] INDEXES } [FOR { TABLE ALL [INDEXED] COLUMNS [SIZE integer] COLUMNS [SIZE integer] { column attribute } [SIZE integer] [{ column attribute } [SIZE integer]]... ALL [LOCAL] INDEXES }]...
for_update_clause	FOR UPDATE [OF [[schema.] { table view } .]column [, [[schema.] { table view } .]column]...] [NOWAIT WAIT integer]

하위 절	구문
full_database_recovery	[STANDBY] DATABASE [{ UNTIL { CANCEL TIME date CHANGE integer } USING BACKUP CONTROLFILE } [UNTIL { CANCEL TIME date CHANGE integer } USING BACKUP CONTROLFILE]...]
fully_qualified_file_name	+diskgroup_name/db_name/file_type/ file_type_tag.filenumbers.incarnation_number
function_association	{ FUNCTIONS schema.]function [, [schema.]function]... PACKAGES [schema.]package [, [schema.]package]... TYPES [schema.]type [, [schema.]type]... INDEXES [schema.]index [, [schema.]index]... INDEXTYPES [schema.]indextype [, [schema.]indextype]... { using_statistics_type { default_cost_clause [, default_selectivity_clause] default_selectivity_clause [, default_cost_clause] } }
function_declaration	FUNCTION name (parameter datatype[, parameter datatype]...) RETURN datatype { IS AS } { pl/sql_block call_spec }
function_spec	FUNCTION name (parameter datatype [, parameter datatype]...) return_clause

하위 절	구문
general_recovery	<pre> RECOVER [AUTOMATIC] [FROM 'location'] { { full_database_recovery partial_database_recovery LOGFILE 'filename' } [{ TEST ALLOW integer CORRUPTION parallel_clause } [TEST ALLOW integer CORRUPTION parallel_clause]...] CONTINUE [DEFAULT] CANCEL } </pre>
global_partitioned_index	<pre> GLOBAL PARTITION BY { RANGE (column_list) (index_partitioning_clause) HASH (column_list) { individual_hash_partitions hash_partitions_by_quantity } } </pre>
grant_object_privileges	<pre> { object_privilege ALL [PRIVILEGES] } [(column [, column]...)] [, { object_privilege ALL [PRIVILEGES] } [(column [, column]...)]]... on_object_clause TO grantee_clause [WITH HIERARCHY OPTION] [WITH GRANT OPTION] </pre>
grant_system_privileges	<pre> { system_privilege role ALL PRIVILEGES } [, { system_privilege role ALL PRIVILEGES }] ... TO grantee_clause [IDENTIFIED BY password] [WITH ADMIN OPTION] </pre>
grantee_clause	<pre> { user role PUBLIC } [, { user role PUBLIC }]... </pre>

하위 절	구문
group_by_clause	<pre>GROUP BY { expr rollup_cube_clause grouping_sets_clause } [, { expr rollup_cube_clause grouping_sets_clause }] ... [HAVING condition]</pre>
grouping_expression_list	expression_list [, expression_list]...
grouping_sets_clause	GROUPING SETS ({ rollup_cube_clause grouping_expression_list })
hash_partitioning	PARTITION BY HASH (column [, column] ...) { individual_hash_partitions hash_partitions_by_quantity }
hash_partitions_by_quantity	PARTITIONS hash_partition_quantity [STORE IN (tablespace [, tablespace]...)] [OVERFLOW STORE IN (tablespace [, tablespace]...)]
hierarchical_query_clause	[START WITH condition] CONNECT BY [NOCYCLE] condition
hierarchy_clause	HIERARCHY hierarchy (child_level CHILD OF parent_level [CHILD OF parent_level]... [dimension_join_clause])
implementation_clause	{ ANCILLARY TO primary_operator (parameter_type [, parameter_type]...) [, primary_operator (parameter_type [, parameter_type [, parameter_type]...)] ... context_clause }
incomplete_file_name	+diskgroup_name [(template_name)]

하위 절	구문
index_attributes	[{ physical_attributes_clause logging_clause ONLINE COMPUTE STATISTICS TABLESPACE { tablespace DEFAULT } key_compression { SORT NOSORT } REVERSE parallel_clause } [physical_attributes_clause logging_clause ONLINE COMPUTE STATISTICS TABLESPACE { tablespace DEFAULT } key_compression { SORT NOSORT } REVERSE parallel_clause]...]
index_expr	{ column column_expression }
index_org_overflow_clause	[INCLUDING column_name] OVERFLOW [segment_attributes_clause]
index_org_table_clause	[{ mapping_table_clause PCTTHRESHOLD integer key_compression } [mapping_table_clause PCTTHRESHOLD integer key_compression]...] [index_org_overflow_clause]
index_partition_description	PARTITION [partition [{ segment_attributes_clause key_compression } [segment_attributes_clause key_compression]...]]
index_partitioning_clause	PARTITION [partition] VALUES LESS THAN (value[, value...]) [segment_attributes_clause]

하위 절	구문
index_properties	[{ { global_partitioned_index local_partitioned_index } index_attributes } [{ { global_partitioned_index local_partitioned_index } index_attributes }]... domain_index_clause]
index_subpartition_clause	{ STORE IN (tablespace[, tablespace]...) (SUBPARTITION [subpartition [TABLESPACE tablespace]] [, SUBPARTITION [subpartition [TABLESPACE tablespace]]]...) }
individual_hash_partitions	(PARTITION [partition partitioning_storage_clause] [, PARTITION [partition partitioning_storage_clause]]...)
inheritance_clauses	[NOT] { OVERRIDING FINAL INSTANTIABLE } [[NOT] { OVERRIDING FINAL INSTANTIABLE }]...
inline_constraint	[CONSTRAINT constraint_name] { [NOT] NULL UNIQUE PRIMARY KEY references_clause CHECK (condition) } [constraint_state]
inline_ref_constraint	{ SCOPE IS [schema.] scope_table WITH ROWID [CONSTRAINT constraint_name] references_clause [constraint_state] }

하위 절	구문
inner_cross_join_clause	<pre>table_reference { [INNER] JOIN table_reference { ON condition USING (column [, column]...) } { CROSS NATURAL [INNER] } JOIN table_reference }</pre>
insert_into_clause	INTO dml_table_expression_clause [t_alias] [(column [, column]...)]
integer	[+ -] digit [digit]...
interval_day_to_second	<pre>INTERVAL ' { integer integer time_expr time_expr } ' { { DAY HOUR MINUTE } [(leading_precision)] SECOND [(leading_precision [, fractional_seconds_precision])] } [TO { DAY HOUR MINUTE SECOND [(fractional_seconds_precision)] }]</pre>
interval_year_to_month	<pre>INTERVAL 'integer [- integer]' { YEAR MONTH } [(precision)] [TO { YEAR MONTH }]</pre>
into_clause	INTO [schema.] table
invoker_rights_clause	AUTHID { CURRENT_USER DEFINER }
Java_declaration	JAVA NAME 'string'
join_clause	{ inner_cross_join_clause outer_join_clause }
key_compression	{ COMPRESS [integer] NOCOMPRESS }
level_clause	<pre>LEVEL level IS { level_table.level_column (level_table.level_column [, level_table.level_column]...) }</pre>

하위 절	구문
list_partitioning	PARTITION BY LIST (column) (PARTITION [partition] list_values_clause table_partition_description [, PARTITION [partition] list_values_clause table_partition_description]...)
list_values_clause	VALUES ({ value NULL [, { value NULL }...] DEFAULT })
LOB_parameters	{ TABLESPACE tablespace { ENABLE DISABLE } STORAGE IN ROW storage_clause CHUNK integer PCTVERSION integer RETENTION FREEPOOLS integer { CACHE { NOCACHE CACHE READS } [logging_clause] } } [TABLESPACE tablespace { ENABLE DISABLE } STORAGE IN ROW storage_clause CHUNK integer PCTVERSION integer RETENTION FREEPOOLS integer { CACHE { NOCACHE CACHE READS } [logging_clause] }] ...
LOB_partition_storage	PARTITION partition { LOB_storage_clause varray_col_properties } [LOB_storage_clause varray_col_properties] ... [(SUBPARTITION subpartition { LOB_storage_clause varray_col_properties } [LOB_storage_clause varray_col_properties]...)]

하위 절	구문
LOB_storage_clause	<pre> LOB { (LOB_item [, LOB_item]...) STORE AS (LOB_parameters) (LOB_item) STORE AS { LOB_segname (LOB_parameters) LOB_segname (LOB_parameters) } } </pre>
local_partitioned_index	<pre> LOCAL [on_range_partitioned_table on_list_partitioned_table on_hash_partitioned_table on_comp_partitioned_table] </pre>
logfile_clause	<pre> LOGFILE [GROUP integer] file_specification [, [GROUP integer] file_specification]... </pre>
logfile_clauses	<pre> { { ARCHIVELOG [MANUAL] NOARCHIVELOG } [NO] FORCE LOGGING RENAME FILE 'filename' [, 'filename']... TO 'filename' CLEAR [UNARCHIVED] LOGFILE logfile_descriptor [, logfile_descriptor]... [UNRECOVERABLE DATAFILE] add_logfile_clauses drop_logfile_clauses supplemental_db_logging } </pre>
logfile_descriptor	<pre> { GROUP integer ('filename' [, 'filename']...) 'filename' } </pre>
logging_clause	{ LOGGING NOLOGGING }
main_model	<pre> [MAIN main_model_name] model_column_clauses [cell_reference_options] model_rules_clause </pre>
managed_standby_recovery	<pre> RECOVER MANAGED STANDBY DATABASE [recover_clause cancel_clause finish_clause] </pre>

하위 절	구문
map_order_func_declaration	{ MAP ORDER } MEMBER function_declaration
map_order_function_spec	{ MAP ORDER } MEMBER function_spec
mapping_table_clauses	{ MAPPING TABLE NOMAPPING }
materialized_view_props	[column_properties] [table_partitioning_clauses] [CACHE NOCACHE] [parallel_clause] [build_clause]
maximize_standby_db_clause	SET STANDBY DATABASE TO MAXIMIZE { PROTECTION AVAILABILITY PERFORMANCE }
maxsize_clause	MAXSIZE { UNLIMITED size_clause }
merge_insert_clause	WHEN NOT MATCHED THEN INSERT [(column [, column]...)] VALUES ({ expr [, expr]... DEFAULT }) [where_clause]
merge_table_partitions	MERGE PARTITIONS partition_1, partition_2 [INTO partition_spec] [update_index_clauses] [parallel_clause]
merge_table_subpartitions	MERGE SUBPARTITIONS subpart_1, subpart_2 [INTO subpartition_spec] [update_index_clauses] [parallel_clause]
merge_update_clause	WHEN MATCHED THEN UPDATE SET column = { expr DEFAULT } [, column = { expr DEFAULT }]... [where_clause] [DELETE where_clause]
model_clause	MODEL [cell_reference_options] [return_rows_clause] [reference_model] [reference_model]... main_model
model_column	expr [[AS] c_alias]
model_column_clauses	[query_partition_clause [c_alias]] DIMENSION BY (model_column [, model_column]...) MEASURES (model_column [, model_column]...)

하위 절	구문
model_rules_clause	<pre> RULES [UPSERT UPDATE] [{ AUTOMATIC SEQUENTIAL } ORDER] [ITERATE (number) [UNTIL (condition)]] ([UPDATE UPSERT] cell_assignment [order_by_clause] = expr [[UPDATE UPSERT] cell_assignment [order_by_clause] = expr]...) </pre>
modify_col_properties	<pre> (column [datatype] [DEFAULT expr] [inline_constraint [inline_constraint]...] [LOB_storage_clause] [, column [datatype] [DEFAULT expr] [inline_constraint [inline_constraint]...] [LOB_storage_clause]]) </pre>
modify_col_substitutable	COLUMN column [NOT] SUBSTITUTABLE AT ALL LEVELS [FORCE]
modify_collection_retrieval	MODIFY NESTED TABLE collection_item RETURN AS { LOCATOR VALUE }
modify_column_clauses	MODIFY { modify_col_properties modify_col_substitutable }
modify_hash_partition	MODIFY PARTITION partition { partition_attributes alter_mapping_table_clause [REBUILD] UNUSABLE LOCAL INDEXES }
modify_hash_subpartition	{ { allocate_extent_clause deallocate_unused_clause shrink_clause { LOB LOB_item VARRAY varray } modify_LOB_parameters [{ LOB LOB_item VARRAY varray } modify_LOB_parameters]... } [REBUILD] UNUSABLE LOCAL INDEXES }

하위 절	구문
modify_index_defaultAttrs	<pre> MODIFY DEFAULT ATTRIBUTES [FOR PARTITION partition] { physical_attributes_clause TABLESPACE { tablespace DEFAULT } logging_clause } [physical_attributes_clause TABLESPACE { tablespace DEFAULT } logging_clause]... </pre>
modify_index_partition	<pre> MODIFY PARTITION partition { { deallocate_unused_clause allocate_extent_clause physical_attributes_clause logging_clause key_compression } [deallocate_unused_clause allocate_extent_clause physical_attributes_clause logging_clause key_compression]... PARAMETERS ('ODCI_parameters') COALESCE UPDATE BLOCK REFERENCES UNUSABLE } </pre>
modify_index_subpartition	<pre> MODIFY SUBPARTITION subpartition { UNUSABLE allocate_extent_clause deallocate_unused_clause } </pre>
modify_list_partition	<pre> MODIFY PARTITION partition { partition_attributes {ADD DROP} VALUES (partition_value[, partition_value]...) [REBUILD] UNUSABLE LOCAL INDEXES } </pre>
modify_list_subpartition	<pre> { allocate_extent_clause deallocate_unused_clause shrink_clause { LOB LOB_item VARRAY varray } modify_LOB_parameters [{ LOB LOB_item VARRAY varray } modify_LOB_parameters] ... [REBUILD] UNUSABLE LOCAL INDEXES { ADD DROP } VALUES (value[, value]...) } </pre>

하위 절	구문
modify_LOB_parameters	<pre> { storage_clause PCTVERSION integer RETENTION FREEPOOLS integer REBUILD FREEPOOLS { CACHE { NOCACHE CACHE READS } [logging_clause] } allocate_extent_clause deallocate_unused_clause shrink_clause } [storage_clause PCTVERSION integer RETENTION FREEPOOLS integer REBUILD FREEPOOLS { CACHE { NOCACHE CACHE READS } [logging_clause] } allocate_extent_clause deallocate_unused_clause shrink_clause]... </pre>
modify_LOB_storage_clause	MODIFY LOB (LOB_item) (modify_LOB_parameters)
modify_range_partition	<pre> MODIFY PARTITION partition { partition_attributes { add_hash_subpartition add_list_subpartition } COALESCE SUBPARTITION [update_index_clauses] [parallel_clause] alter_mapping_table_clause [REBUILD] UNUSABLE LOCAL INDEXES } </pre>
modify_table_defaultAttrs	<pre> MODIFY DEFAULT ATTRIBUTES [FOR PARTITION partition] [segment_attributes_clause] [table_compression] [PCTTHRESHOLD integer] [key_compression] [alter_overflow_clause] [{ LOB (LOB_item) VARRAY varray } (LOB_parameters) [{ LOB (LOB_item) VARRAY varray } (LOB_parameters)]...] </pre>

하위 절	구문
modify_table_partition	{ modify_range_partition modify_hash_partition modify_list_partition }
modify_table_subpartition	MODIFY SUBPARTITION subpartition { modify_hash_subpartition modify_list_subpartition }
move_table_clause	MOVE [ONLINE] [segment_attributes_clause] [table_compression] [index_org_table_clause] [{ LOB_storage_clause varray_col_properties } [{ LOB_storage_clause varray_col_properties }]...] [parallel_clause]
move_table_partition	MOVE PARTITION partition [MAPPING TABLE] [table_partition_description] [update_index_clauses] [parallel_clause]
move_table_subpartition	MOVE SUBPARTITION subpartition_spec [update_index_clauses] [parallel_clause]
multi_column_for_loop	FOR (dimension_column [, dimension_column]...) IN ({ (literal [, literal]...) [(literal [, literal]...)...] subquery })
multi_table_insert	{ ALL insert_into_clause [values_clause] [insert_into_clause [values_clause]]... conditional_insert_clause } subquery
multiset_except	nested_table1 MULTISET EXCEPT [ALL DISTINCT] nested_table2

하위 절	구문
multiset_intersect	nested_table1 MULTISET INTERSECT [ALL DISTINCT] nested_table2
multiset_union	nested_table1 MULTISET UNION [ALL DISTINCT] nested_table2
nested_table_col_properties	NESTED TABLE { nested_item COLUMN_VALUE } [substitutable_column_clause] STORE AS storage_table [({ object_properties) [physical_properties] [column_properties] } [(object_properties) [physical_properties] [column_properties]]...)] [RETURN AS { LOCATOR VALUE }]
new_values_clause	{ INCLUDING EXCLUDING } NEW VALUES
number	[+ -] { digit [digit]... [.] [digit [digit]...] . digit [digit]... } [e [+ -] digit [digit]... [f d]
numeric_file_name	+diskgroup_name.filenumbers.incarnation_number
object_properties	{ { column attribute } [DEFAULT expr] [inline_constraint [inline_constraint]... inline_ref_constraint] { out_of_line_constraint out_of_line_ref_constraint supplemental_logging_props }
object_table	CREATE [GLOBAL TEMPORARY] TABLE [schema.]table OF [schema.]object_type [object_table_substitution] [(object_properties)] [ON COMMIT { DELETE PRESERVE } ROWS] [OID_clause] [OID_index_clause] [physical_properties] [table_properties] ;

하위 절	구문
object_table_substitution	[NOT] SUBSTITUTABLE AT ALL LEVELS
object_type_col_properties	COLUMN column substitutable_column_clause
object_view_clause	<pre> OF [schema.]type_name { WITH OBJECT IDENTIFIER { DEFAULT (attribute [, attribute]...) } UNDER [schema.]superview } ({ out_of_line_constraint attribute inline_constraint [inline_constraint]... } [, { out_of_line_constraint attribute inline_constraint [inline_constraint]... }])...) </pre>
OID_clause	OBJECT IDENTIFIER IS { SYSTEM GENERATED PRIMARY KEY }
OID_index_clause	<pre> OIDINDEX [index] ({ physical_attributes_clause TABLESPACE tablespace } [physical_attributes_clause TABLESPACE tablespace]...) </pre>
on_comp_partitioned_table	<pre> [STORE IN (tablespace [, tablespace]...)] (PARTITION [partition [{ segment_attribute_clause key_compression } [segment_attribute_clause key_compression]...] [index_subpartition_clause]] [, PARTITION [partition [{ segment_attribute_clause key_compression } [segment_attribute_clause key_compression]...] [index_subpartition_clause]]...) </pre>

하위 절	구문
on_hash_partitioned_table	{ STORE IN (tablespace[, tablespace]...) (PARTITION [partition [TABLESPACE tablespace]] [, PARTITION [partition [TABLESPACE tablespace]]]...) }
on_list_partitioned_table	(PARTITION [partition [{ segment_attributes_clause key_compression }] [{ segment_attributes_clause key_compression }]...]] [, PARTITION [partition [{ segment_attributes_clause key_compression }] [{ segment_attributes_clause key_compression }]...]])...
on_object_clause	{ schema.object { DIRECTORY directory_name JAVA { SOURCE RESOURCE } [schema.]object } }
on_range_partitioned_table	(PARTITION [partition [{ segment_attributes_clause key_compression }] [{ segment_attributes_clause key_compression }]...]] [, PARTITION [partition [{ segment_attributes_clause key_compression }] [{ segment_attributes_clause key_compression }]...]])...

하위 절	구문								
order_by_clause	<pre> ORDER [SIBLINGS] BY { expr position c_alias } [ASC DESC] [NULLS FIRST NULLS LAST] [, { expr position c_alias } [ASC DESC] [NULLS FIRST NULLS LAST]]... </pre>								
out_of_line_constraint	<pre> [CONSTRAINT constraint_name] { UNIQUE (column [, column]...) PRIMARY KEY (column [, column]...) FOREIGN KEY (column [, column]...) references_clause CHECK (condition) } [constraint_state] </pre>								
out_of_line_ref_constraint	<pre> { SCOPE FOR ({ ref_col ref_attr }) IS [schema.]scope_table REF ({ ref_col ref_attr }) WITH ROWID [CONSTRAINT constraint_name] FOREIGN KEY ({ ref_col ref_attr }) references_clause [constraint_state] } </pre>								
outer_join_clause	<table> <tr> <td>table_reference</td> </tr> <tr> <td>[query_partition_clause]</td> </tr> <tr> <td>{ outer_join_type JOIN</td> </tr> <tr> <td> NATURAL [outer_join_type] JOIN</td> </tr> <tr> <td>}</td> </tr> <tr> <td>table_reference [query_partition_clause]</td> </tr> <tr> <td>[ON condition</td> </tr> <tr> <td> USING (column [, column]...)</td> </tr> </table>	table_reference	[query_partition_clause]	{ outer_join_type JOIN	NATURAL [outer_join_type] JOIN	}	table_reference [query_partition_clause]	[ON condition	USING (column [, column]...)
table_reference									
[query_partition_clause]									
{ outer_join_type JOIN									
NATURAL [outer_join_type] JOIN									
}									
table_reference [query_partition_clause]									
[ON condition									
USING (column [, column]...)									
outer_join_type	<table> <tr> <td>{ FULL LEFT RIGHT }</td> </tr> <tr> <td>[OUTER]</td> </tr> </table>	{ FULL LEFT RIGHT }	[OUTER]						
{ FULL LEFT RIGHT }									
[OUTER]									
parallel_clause	<table> <tr> <td>{ NOPARALLEL PARALLEL [integer] }</td> </tr> </table>	{ NOPARALLEL PARALLEL [integer] }							
{ NOPARALLEL PARALLEL [integer] }									
parallel_enable_clause	<table> <tr> <td>PARALLEL_ENABLE</td> </tr> <tr> <td>[(PARTITION argument BY { ANY { HASH RANGE } (column [, column]...) }) [streaming_clause]]</td> </tr> </table>	PARALLEL_ENABLE	[(PARTITION argument BY { ANY { HASH RANGE } (column [, column]...) }) [streaming_clause]]						
PARALLEL_ENABLE									
[(PARTITION argument BY { ANY { HASH RANGE } (column [, column]...) }) [streaming_clause]]									

하위 절	구문
partial_database_recovery	{ TABLESPACE tablespace [, tablespace]... DATAFILE { 'filename' filenumber } [, 'filename' filenumber]... } STANDBY { TABLESPACE tablespace [, tablespace]... DATAFILE { 'filename' filenumber } [, 'filename' filenumber]... } } UNTIL [CONSISTENT WITH] CONTROLFILE }
partition_attributes	[{ physical_attributes_clause logging_clause allocate_extent_clause deallocate_unused_clause shrink_clause } [physical_attributes_clause logging_clause allocate_extent_clause deallocate_unused_clause shrink_clause]...] [OVERFLOW { physical_attributes_clause logging_clause allocate_extent_clause deallocate_unused_clause } [physical_attributes_clause logging_clause allocate_extent_clause deallocate_unused_clause]...] [table_compression] [{ LOB LOB_item VARRAY varray } modify_LOB_parameters [{ LOB LOB_item VARRAY varray } modify_LOB_parameters]...]
partition_extended_name	[schema.] { table view } [PARTITION (partition) SUBPARTITION (subpartition)]
partition_level_subpartition	{ SUBPARTITIONS hash_subpartition_quantity [STORE IN (tablespace[, tablespace]...)] (subpartition_spec[, subpartition_spec]...) }

하위 절	구문
partition_spec	PARTITION [partition] [table_partition_description]
partitioning_storage_clause	[{ TABLESPACE tablespace OVERFLOW [TABLESPACE tablespace] LOB (LOB_item) STORE AS { LOB_segnane [(TABLESPACE tablespace)] (TABLESPACE tablespace) } VARRAY varray_item STORE AS LOB LOB_segnane } [{ TABLESPACE tablespace OVERFLOW [TABLESPACE tablespace] LOB (LOB_item) STORE AS { LOB_segnane [(TABLESPACE tablespace)] (TABLESPACE tablespace) } VARRAY varray_item STORE AS LOB LOB_segnane }]]...
password_parameters	{ { FAILED_LOGIN_ATTEMPTS PASSWORD_LIFE_TIME PASSWORD_REUSE_TIME PASSWORD_REUSE_MAX PASSWORD_LOCK_TIME PASSWORD_GRACE_TIME } { expr UNLIMITED DEFAULT } PASSWORD_VERIFY_FUNCTION { function NULL DEFAULT } }
permanent_tablespace_clause	{ MINIMUM EXTENT integer [K M] BLOCKSIZE integer [K] logging_clause FORCE LOGGING DEFAULT [table_compression] storage_clause { ONLINE OFFLINE } extent_management_clause segment_management_clause flashback_mode_clause MINIMUM EXTENT integer [K M] BLOCKSIZE integer [K] logging_clause FORCE LOGGING DEFAULT [table_compression] storage_clause { ONLINE OFFLINE } extent_management_clause segment_management_clause flashback_mode_clause] }

하위 절	구문
physical_attributes_clause	[{ PCTFREE integer PCTUSED integer INITTRANS integer storage_clause } [PCTFREE integer PCTUSED integer INITTRANS integer storage_clause] ...]
physical_properties	{ segment_attributes_clause [table_compression] ORGANIZATION { HEAP [segment_attributes_clause] [table_compression] INDEX [segment_attributes_clause] index_org_table_clause EXTERNAL external_table_clause } CLUSTER cluster (column [, column]...) }
pragma_clause	PRAGMA RESTRICT_REFERENCES ({ method_name DEFAULT } , { RNDS WNDS RNPS WNPS TRUST } [, { RNDS WNDS RNPS WNPS TRUST }]...)
procedure_declaration	PROCEDURE name (parameter datatype [, parameter datatype]...) { IS AS } { pl/sql_block call_spec }
procedure_spec	PROCEDURE name (parameter datatype [, parameter datatype]...) [{ IS AS } call_spec]
proxy_authentication	{ AUTHENTICATION REQUIRED AUTHENTICATED USING { PASSWORD DISTINGUISHED NAME CERTIFICATE [TYPE 'type'] [VERSION 'version'] } }

하위 절	구문
proxy_clause	<pre>{ GRANT REVOKE } CONNECT THROUGH proxy [WITH { ROLE { role_name [, role_name]... ALL EXCEPT role_name [, role_name]... } NO ROLES }] [proxy_authentication]</pre>
qualified_disk_clause	<pre>search_string [NAME disk_name] [SIZE size_clause] [FORCE NOFORCE]</pre>
qualified_template_clause	<pre>template_name ATTRIBUTES ([MIRROR UNPROTECTED] [FINE COARSE])</pre>
query_partition_clause	<pre>PARTITION BY { value_expr[, value_expr]... (value_expr[, value_expr]...)}</pre>
query_table_expression	<pre>{ query_name [schema.] { table [{ PARTITION (partition) SUBPARTITION (subpartition) } [sample_clause] [sample_clause] @ dblink] { view materialized view } [@ dblink] } (subquery [subquery_restriction_clause]) table_collection_expression }</pre>
quiesce_clauses	QUIESCE RESTRICTED UNQUIESCE
range_partitioning	<pre>PARTITION BY RANGE (column[, column]...) (PARTITION [partition] range_values_clause table_partition_description [, PARTITION [partition] range_values_clause table_partition_description]...)</pre>

하위 절	구문
range_values_clause	VALUES LESS THAN ({ value MAXVALUE } [, { value MAXVALUE }]...)
rebalance_diskgroup_clause	REBALANCE [POWER integer]
rebuild_clause	REBUILD [{ PARTITION partition SUBPARTITION subpartition } { REVERSE NOREVERSE }] [parallel_clause TABLESPACE tablespace PARAMETERS ('ODCI_parameters') ONLINE COMPUTE STATISTICS physical_attributes_clause key_compression logging_clause] [parallel_clause TABLESPACE tablespace PARAMETERS ('ODCI_parameters') ONLINE COMPUTE STATISTICS physical_attributes_clause key_compression logging_clause]... ...
records_per_block_clause	{ MINIMIZE NOMINIMIZE } RECORDS_PER_BLOCK
recover_clause	{ { DISCONNECT [FROM SESSION] { TIMEOUT integer NOTIMEOUT } } { NODELAY DEFAULT DELAY DELAY integer } NEXT integer { EXPIRE integer NO EXPIRE } parallel_clause USING CURRENT LOGFILE UNTIL CHANGE integer THROUGH { [THREAD integer] SEQUENCE integer ALL ARCHIVELOG { ALL LAST NEXT } SWITCHOVER } } [{ DISCONNECT [FROM SESSION] { TIMEOUT integer NOTIMEOUT } } { NODELAY DEFAULT DELAY DELAY integer } NEXT integer { EXPIRE integer NO EXPIRE } parallel_clause USING CURRENT LOGFILE

하위 절	구문
	<pre> UNTIL CHANGE integer THROUGH { [THREAD integer] SEQUENCE integer ALL ARCHIVELOG { ALL LAST NEXT } SWITCHOVER } ... </pre>
recovery_clauses	<pre> { general_recovery managed_standby_recovery BEGIN BACKUP END BACKUP } </pre>
redo_log_file_spec	<pre> ['filename' ('filename' [, 'filename']...)] [SIZE size_clause] [REUSE] </pre>
redo_thread_clauses	<pre> { ENABLE DISABLE } { INSTANCE 'instance_name' [PUBLIC] THREAD integer } </pre>
reference_model	<pre> REFERENCE reference_spreadsheet_name ON (subquery) spreadsheet_column_clauses [cell_reference_options] </pre>
references_clause	<pre> REFERENCES [schema.] { object_table view } [(column [, column]...)] [ON DELETE { CASCADE SET NULL }] [constraint_state] </pre>
referencing_clause	<pre> REFERENCING { OLD [AS] old NEW [AS] new PARENT [AS] parent } [OLD [AS] old NEW [AS] new PARENT [AS] parent]... </pre>
register_logfile_clause	<pre> REGISTER [OR REPLACE] [PHYSICAL LOGICAL] LOGFILE [file_specification [, file_specification]...] FOR logminer_session_name </pre>

하위 절	구문
relational_properties	<pre> { column datatype [SORT] [DEFAULT expr] [inline_constraint [inline_constraint]... inline_ref_constraint] { out_of_line_constraint out_of_line_ref_constraint supplemental_logging_props } } [, { column datatype [SORT] [DEFAULT expr] [inline_constraint [inline_constraint]... inline_ref_constraint] { out_of_line_constraint out_of_line_ref_constraint supplemental_logging_props }] ...]</pre>
relational_table	<pre> CREATE [GLOBAL TEMPORARY] TABLE [schema.]table [(relational_properties)] [ON COMMIT { DELETE PRESERVE } ROWS] [physical_properties] [table_properties] ;</pre>
rename_column_clause	RENAME COLUMN old_name TO new_name
rename_index_partition	<pre> RENAME { PARTITION partition SUBPARTITION subpartition } TO new_name</pre>
rename_partition_subpart	<pre> RENAME { PARTITION SUBPARTITION } current_name TO new_name</pre>
replace_type_clause	<pre> REPLACE [invoker_rights_clause] AS OBJECT (attribute datatype [, attribute datatype] ... [, element_spec [, element_spec]...])</pre>
resize_disk_clauses	<pre> RESIZE { ALL [SIZE size_clause] DISK disk_name [SIZE size_clause] [, disk_name [SIZE size_clause]]... DISKS IN FAILGROUP failgroup_name [SIZE size_clause] [, failgroup_name [SIZE size_clause]]... }</pre>

하위 절	구문
resource_parameters	{ { SESSIONS_PER_USER CPU_PER_SESSION CPU_PER_CALL CONNECT_TIME IDLE_TIME LOGICAL_READS_PER_SESSION LOGICAL_READS_PER_CALL COMPOSITE_LIMIT } { integer UNLIMITED DEFAULT } PRIVATE_SGA { integer [K M] UNLIMITED DEFAULT } }
restricted_session_clauses	{ ENABLE DISABLE } RESTRICTED SESSION
return_clause	{ RETURN datatype [{ IS AS } call_spec] sqlj_object_type_sig }
return_rows_clause	RETURN { UPDATED ALL } ROWS
returning_clause	RETURNING expr [, expr]... INTO data_item [, data_item]...
revoke_object_privileges	{ object_privilege ALL [PRIVILEGES] } [, { object_privilege ALL [PRIVILEGES] }]... on_object_clause FROM grantee_clause [CASCADE CONSTRAINTS FORCE]
revoke_system_privileges	{ system_privilege role ALL PRIVILEGES } [, { system_privilege role ALL PRIVILEGES }]... FROM grantee_clause
rollup_cube_clause	{ ROLLUP CUBE } (grouping_expression_list)
routine_clause	[schema.] [type. package.] { function procedure method } [@dblink_name] ([argument [, argument]...])
row_movement_clause	{ ENABLE DISABLE } ROW MOVEMENT

하위 절	구문
sample_clause	SAMPLE [BLOCK] (sample_percent) [SEED (seed_value)]
schema_object_clause	{ object_option [, object_option]... ALL } auditing_on_clause
scoped_table_ref_constraint	{ SCOPE FOR ({ ref_column ref_attribute }) IS [schema.] { scope_table_name c_alias } } [, SCOPE FOR ({ ref_column ref_attribute }) IS [schema.] { scope_table_name c_alias } }]...
searched_case_expression	WHEN condition THEN return_expr [WHEN condition THEN return_expr]...
security_clause	GUARD { ALL STANDBY NONE }
segment_attributes_clause	{ physical_attributes_clause TABLESPACE tablespace logging_clause } [physical_attributes_clause TABLESPACE tablespace logging_clause]...
segment_management_clause	SEGMENT SPACE MANAGEMENT { MANUAL AUTO }
select_list	{ * { query_name.* [schema.] { table view materialized view } .* expr [[AS] c_alias] } [, { query_name.* [schema.] { table view materialized view } .* expr [[AS] c_alias] }]... }

하위 절	구문
set_subpartition_template	<pre>SET SUBPARTITION TEMPLATE { (SUBPARTITION subpartition [list_values_clause] [partitioning_storage_clause] [, SUBPARTITION subpartition [list_values_clause] [partitioning_storage_clause]...]) hash_subpartition_quantity }</pre>
set_time_zone_clause	SET TIME_ZONE = '{ { + - } hh : mi time_zone_region }'
shrink_clause	SHRINK SPACE [COMPACT] [CASCADE]
shutdown_dispatcher_clause	SHUTDOWN [IMMEDIATE] dispatcher_name
simple_case_expression	<pre>expr WHEN comparison_expr THEN return_expr [WHEN comparison_expr THEN return_expr]...</pre>
single_column_for_loop	<pre>FOR dimension_column { IN ({ literal [, literal]... subquery }) [LIKE pattern] FROM literal TO literal { INCREMENT DECREMENT } literal }</pre>
single_table_insert	<pre>insert_into_clause { values_clause [returning_clause] subquery }</pre>
size_clause	integer [K M G T]
split_index_partition	<pre>SPLIT PARTITION partition_name_old AT (value [, value]...) [INTO (index_partition_description, index_partition_description)] [parallel_clause]</pre>
split_table_partition	<pre>SPLIT PARTITION current_partition { AT VALUES } (value [, value]...) [INTO (partition_spec, partition_spec)] [update_index_clauses] [parallel_clause]</pre>

하위 절	구문
split_table_subpartition	<pre>SPLIT SUBPARTITION subpartition VALUES ({ value NULL } [, value NULL]...) [INTO (subpartition_spec, subpartition_spec)] [update_index_clauses] [parallel_clause]</pre>
sql_statement_clause	<pre>{ { statement_option ALL } [, { statement_option ALL }]... { system_privilege ALL PRIVILEGES } [, { system_privilege ALL PRIVILEGES }]... } [auditing_by_clause]</pre>
sqlj_object_type	EXTERNAL NAME java_ext_name LANGUAGE JAVA USING (SQLData CustomDatum OraData)
sqlj_object_type_attr	EXTERNAL NAME 'field_name'
sqlj_object_type_sig	<pre>RETURN { datatype SELF AS RESULT } EXTERNAL { VARIABLE NAME 'java_static_field_name' NAME 'java_method_sig' }</pre>
standby_database_clauses	<pre>(activate_standby_db_clause maximize_standby_db_clause register_logfile_clause commit_switchover_clause start_standby_clause stop_standby_clause) [parallel_clause]</pre>
start_standby_clause	<pre>START LOGICAL STANDBY APPLY [IMMEDIATE] [NODELAY] [NEW PRIMARY dblink INITIAL [scn_value] { SKIP FAILED TRANSACTION FINISH }]</pre>
startup_clauses	<pre>{ MOUNT [{ STANDBY CLONE } DATABASE] OPEN { [READ WRITE] [RESETLOGS NORESETLOGS] [UPGRADE DOWNGRADE] READ ONLY } }</pre>
stop_standby_clause	<pre>{ STOP ABORT } LOGICAL STANDBY APPLY</pre>

하위 절	구문
storage_clause	<pre> STORAGE ({ INITIAL integer [K M] NEXT integer [K M] MINEXTENTS integer MAXEXTENTS { integer UNLIMITED } PCTINCREASE integer FREELISTS integer FREELIST GROUPS integer OPTIMAL [integer [K M] NULL] } BUFFER_POOL { KEEP RECYCLE DEFAULT } } [INITIAL integer [K M] NEXT integer [K M] MINEXTENTS integer MAXEXTENTS { integer UNLIMITED } PCTINCREASE integer FREELISTS integer FREELIST GROUPS integer OPTIMAL [integer [K M] NULL] BUFFER_POOL { KEEP RECYCLE DEFAULT }] ...) </pre>
streaming_clause	{ ORDER CLUSTER } BY (column [, column]...)
subpartition_by_hash	<pre> SUBPARTITION BY HASH (column [, column]...) [SUBPARTITIONS quantity [STORE IN (tablespace [, tablespace]]...) subpartition_template] </pre>
subpartition_by_list	SUBPARTITION BY LIST (column) [subpartition_template]
subpartition_spec	SUBPARTITION [subpartition] [list_values_clause] [partitioning_storage_clause]
subpartition_template	<pre> SUBPARTITION TEMPLATE (SUBPARTITION subpartition [list_values_clause] [partitioning_storage_clause] [, SUBPARTITION subpartition [list_values_clause] [partitioning_storage_clause]]) hash_subpartition_quantity </pre>

하위 절	구문
subprogram_declaration	{ MEMBER STATIC } { procedure_declarati function_declarati constructor_declarati }
subprogram_spec	{ MEMBER STATIC } { procedure_spec function_spec }
subquery	[subquery_factoring_clause] SELECT [hint] [{ { DISTINCT UNIQUE } ALL }] select_list FROM table_reference [, table_reference]... [where_clause] [hierarchical_query_clause] [group_by_clause] [HAVING condition] [model_clause] [{ UNION [ALL] INTERSECT MINUS } (subquery)] [order_by_clause]
subquery_factoring_clause	WITH query_name AS (subquery) [, query_name AS (subquery)]...
subquery_restriction_clause	WITH { READ ONLY CHECK OPTION [CONSTRAINT constraint] }
substitutable_column_clause	[ELEMENT] IS OF [TYPE] ([ONLY] type) [NOT] SUBSTITUTABLE AT ALL LEVELS
supplemental_db_logging	{ ADD DROP } SUPPLEMENTAL LOG { DATA supplemental_id_key_clause }

하위 절	구문
supplemental_id_key_clause	<pre> DATA ({ ALL PRIMARY KEY UNIQUE FOREIGN KEY } [, { ALL PRIMARY KEY UNIQUE FOREIGN KEY }] ...) COLUMNS </pre>
supplemental_log_grp_clause	<pre> GROUP log_group (column [NO LOG] [, column [NO LOG]]...) [ALWAYS] </pre>
supplemental_logging_props	<pre> { supplemental_log_grp_clause supplemental_id_key_clause } </pre>
supplemental_table_logging	<pre> { ADD SUPPLEMENTAL LOG { supplemental_log_grp_clause supplemental_id_key_clause } [, SUPPLEMENTAL LOG { supplemental_log_grp_clause supplemental_id_key_clause }] ... DROP SUPPLEMENTAL LOG { supplemental_id_key_clause GROUP log_group } [, SUPPLEMENTAL LOG { supplemental_id_key_clause GROUP log_group }] } </pre>
table_collection_expression	TABLE (collection_expression) [(+)]
table_compression	{ COMPRESS NOCOMPRESS }
table_index_clause	[schema.]table [t_alias] (index_expr [ASC DESC] [, index_expr [ASC DESC]]...) [index_properties]

하위 절	구문
table_partition_description	[segment_attributes_clause] [table_compression key_compression] [OVERFLOW [segment_attributes_clause]] [{ LOB_storage_clause varray_col_properties } [LOB_storage_clause varray_col_properties]...] [partition_level_subpartition]
table_partitioning_clauses	{ range_partitioning hash_partitioning list_partitioning composite_partitioning }
table_properties	[column_properties] [table_partitioning_clauses] [CACHE NOCACHE] [parallel_clause] [ROWDEPENDENCIES NOROWDEPENDENCIES] [enable_disable_clause] [enable_disable_clause]... [row_movement_clause] [AS subquery]
table_reference	{ ONLY (query_table_expression) [flashback_query_clause] [t_alias] query_table_expression [flashback_query_clause] [t_alias] (join_clause) join_clause }
tablespace_clauses	{ EXTENT MANAGEMENT LOCAL DATAFILE file_specification [, file_specification]... SYSAUX DATAFILE file_specification [, file_specification]... default_tablespace default_temp_tablespace undo_tablespace }
tablespace_group_clause	TABLESPACE GROUP { tablespace_group_name '' }
tablespace_logging_clauses	{ logging_clause [NO] FORCE LOGGING }

하위 절	구문
tablespace_retention_clause	RETENTION { GUARANTEE NOGUARANTEE }
tablespace_state_clauses	{ ONLINE OFFLINE [NORMAL TEMPORARY IMMEDIATE] } READ { ONLY WRITE } { PERMANENT TEMPORARY }
temporary_tablespace_clause	TEMPORARY TABLESPACE tablespace [TEMPFILE file_specification [, file_specification]...] [tablespace_group_clause] [extent_management_clause]
text	[N n] { 'c' [c]...' { Q q } 'quote_delimiter c [c]... quote_delimiter' }
trace_file_clause	TRACE [AS 'filename' [REUSE]] [RESETLOGS NORESETLOGS]
truncate_partition_subpart	TRUNCATE { PARTITION partition SUBPARTITION subpartition } [{ DROP REUSE } STORAGE] [update_index_clauses [parallel_clause]]
undo_tablespace	[BIGFILE SMALLFILE] UNDO TABLESPACE tablespace [TABLESPACE file_specification [, file_specification]...]
undo_tablespace_clause	UNDO TABLESPACE tablespace [DATAFILE file_specification [, file_specification]...] [extent_management_clause] [tablespace_retention_clause]
undrop_disk_clause	UNDROP DISKS

하위 절	구문
update_all_indexes_clause	<pre> UPDATE INDEXES [(index ({ update_index_partition update_index_subpartition })) , (index ({ update_index_partition update_index_subpartition }))] ... </pre>
update_global_index_clause	{ UPDATE INVALIDATE } GLOBAL INDEXES
update_index_clauses	{ update_global_index_clause update_all_indexes_clause }
update_index_partition	<pre> PARTITION [partition] [index_partition_description [index_subpartition_clause]] [, PARTITION [partition] [index_partition_description [index_subpartition_clause]]] ... </pre>
update_index_subpartition	<pre> SUBPARTITION [subpartition] [TABLESPACE tablespace] [, SUBPARTITION [subpartition] [TABLESPACE tablespace]] ... </pre>
update_set_clause	<pre> SET { { (column [, column]...) = (subquery) column = { expr (subquery) DEFAULT } } [, { (column [, column]...) = (subquery) column = { expr (subquery) DEFAULT } }] ... VALUE (t_alias) = { expr (subquery) } } </pre>
upgrade_table_clause	UPGRADE [[NOT] INCLUDING DATA] [column_properties]
using_function_clause	USING [schema.] [package. type.] function_name

하위 절	구문
using_index_clause	USING INDEX { [schema.] index (create_index_statement) index_properties }
using_statistics_type	USING { [schema.] statistics_type NULL }
using_type_clause	USING [schema.] implementation_type [array_DML_clause]
validation_clauses	{ VALIDATE REF UPDATE [SET DANGLING TO NULL] VALIDATE STRUCTURE [CASCADE] [into_clause] { OFFLINE ONLINE } }
values_clause	VALUES ({ expr DEFAULT } [, { expr DEFAULT }] ...)
varray_col_properties	VARRAY varray_item { [substitutable_column_clause] STORE AS LOB { [LOB_segname] (LOB_parameters) LOB_segname } substitutable_column_clause }
where_clause	WHERE condition
windowing_clause	{ ROWS RANGE } { BETWEEN { UNBOUNDED PRECEDING CURRENT ROW value_expr { PRECEDING FOLLOWING } } AND { UNBOUNDED FOLLOWING CURRENT ROW value_expr { PRECEDING FOLLOWING } } { UNBOUNDED PRECEDING CURRENT ROW value_expr PRECEDING } }

하위 절	구문
XML_attributes_clause	XMLATTRIBUTES (value_expr [AS c_alias] [, value_expr [AS c_alias]...])
XMLSchema_spec	[XMLSCHEMA XMLSchema_URL] ELEMENT { element XMLSchema_URL # element }
XMLType_column_properties	XMLTYPE [COLUMN] column [XMLType_storage] [XMLSchema_spec]
XMLType_storage	STORE AS { OBJECT RELATIONAL CLOB [{ LOB_segnane [(LOB_parameters)] LOB_parameters }]
XMLType_table	CREATE TABLE [GLOBAL TEMPORARY] TABLE [schema.]table OF XMLTYPE [(object_properties)] [XMLTYPE XMLType_storage] [XMLSchema_spec] [ON COMMIT { DELETE PRESERVE } ROWS] [OID_clause] [OID_index_clause] [physical_properties] [table_properties] ;
XMLType_view_clause	OF XMLTYPE [XMLSchema_spec] WITH OBJECT IDENTIFIER { DEFAULT (expr [, expr]...) }

부록 D

**Oracle 백그라운드
프로세스**

Oracle Internal & Oracle Academy
Use Only

Oracle 백그라운드 프로세스

이 부록은 모든 백그라운드 프로세스에 대한 리스트가 아니며, 빠른 참조용으로 제작되었습니다. 이러한 백그라운드 프로세스에 대한 자세한 내용 및 여기에서 언급되지 않은 내용은 Oracle Database Reference 설명서를 참조하십시오.

일반 프로세스

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
ARCn	아카이버 프로세스	채워진 리두 로그를 아카이브 로그 위치에 기록합니다. 가능한 프로세스로는 ARC0–ARC9 및 ARCa–ARCb가 있습니다.	아니오	아니오
CJQ0	작업 큐 조정자 프로세스	슬레이브 프로세스(Jnnn)를 생성하여 큐에 있는 작업을 실행합니다.	아니오	예
CKPT	체크포인트 프로세스	콘트롤 파일 및 데이터 파일 헤더에 체크포인트 정보를 기록합니다.	예	예
DBRM	Database Resource Manager 프로세스	리소스 계획을 설정하고 기타 리소스 관리자 작업을 수행합니다.	아니오	예
DBWn	데이터베이스 기록자 프로세스	버퍼 캐시의 더티 버퍼를 데이터 파일에 기록합니다. 가능한 프로세스로는 DBW0–DBW9 및 DBWa–DBWz가 있습니다.	예	예
DIA0	진단 프로세스	정지 탐지 및 deadlock 해결을 담당합니다. 진단 작업을 수행하기 위해 DIAG를 트리거 합니다.	예	예
DIAG	진단 캡처 프로세스	진단 덤프를 수행하고 global oradebug 명령을 실행합니다.	예	예
Dnnn	디스패처 프로세스	Shared Server 구성에서 디스패처는 연결 요청을 연결 요청 큐에 배치합니다.	아니오	예
EMNC	EMON 조정자 프로세스	Streams Event Notification, Continuous Query Notification 및 Fast Application Notification을 포함하여 데이터베이스의 이벤트 관리 및 알림 작업을 조정합니다.	아니오	아니오

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
FBDA	Flashback 데이터 아카이브 프로세스	추적된 테이블에 대한 기록 행을 flashback 데이터 아카이브에 아카이브하고 flashback 아카이브를 관리합니다.	아니오	예
FMON	파일 매핑 모니터 프로세스	저장 영역 공급업체에서 제공되는 매핑 라이브러리와 통신하는 외부 비오라클 데이터베이스 프로세스인 FMPUTL을 생성합니다. 매핑 정보를 관리합니다.	아니오	아니오
Jnnn	작업 큐 슬래이브 프로세스	큐에 있는 작업을 처리합니다. CJQ0에 의해 생성됩니다.	아니오	예
LGWR	로그 기록자 프로세스	로그 베퍼를 리두 로그에 기록합니다.	예	예
MMAN	메모리 관리자 프로세스	SGA 메모리 Broker 기능을 수행하며 메모리 구성 요소의 크기 조정을 담당합니다.	아니오	예
MMNL	Manageability Monitor Lite 프로세스	세션 기록, 캡처 및 metrics 계산과 같은 자주 수행되는 경량 관리 효율성 관련 작업을 수행합니다.	아니오	예
MMON	관리 효율성 모니터 프로세스	Automatic Workload Repository에 대한 통계를 수집합니다.	아니오	예
PMON	프로세스 모니터	실패한 프로세스 리소스를 Recovery합니다. Shared Server 구조가 사용된 경우 PMON은 모든 실패한 디스패처 또는 서버 프로세스를 모니터하고 재시작합니다.	예	예
Pnnn	Parallel Query 슬래이브 프로세스	parallel query 작업에 참여하기 위해 필요에 따라 시작 및 정지됩니다.	아니오	아니오
PSP0	Process Spawner 프로세스	Oracle 프로세스를 시작 및 정지합니다. ASM 리밸런스 슬래이브를 시작/정지하여 RBAL의 작업 로드를 줄입니다.	아니오	예

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
RCBG	결과 캐시 백그라운드 프로세스	SQL query 및 PL/SQL 함수 결과 캐시를 지원합니다.	아니오	아니오
RECO	복구자 프로세스	분산된 트랜잭션과 관련된 failure를 해결합니다.	아니오	예
RVWR	Recovery 기록자 프로세스	Flash Recovery Area의 flashback 데이터베이스 로그에 flashback 데이터를 기록합니다.	아니오	아니오
SMCO	공간 관리 조정자 프로세스	사전 공간 할당(proactive space allocation) 및 공간 회수와 같은 여러 공간 관련 작업의 실행을 조정합니다.	아니오	예
SMON	시스템 모니터 프로세스	Instance failure 이후 Recovery를 수행하고 임시 세그먼트 및 Extent를 모니터합니다.	예	예
Snnn	Shared Server 프로세스	Shared Server 구성에서 Shared Server는 연결 요청 큐(디스패처로 채워진)를 검사하고 연결 요청을 처리합니다.	아니오	예
VKTM	Virtual Keeper of Time 프로세스	실제 시간(매초 갱신됨) 및 참조 시간 카운터(20밀리초마다 갱신되고 상승된 우선 순위에서만 사용할 수 있음)를 제공합니다.	예	예
Wnnn	공간 관리 슬레이브 프로세스	이러한 프로세스는 공간 관리 작업을 실행하기 위해 SMCO가 생성한 슬레이브 프로세스입니다.	아니오	예

고급 큐 처리 프로세스

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
Qnnn	AQ 서버 클래스 프로세스	Streams AQ 큐에 있는 메시지를 처리합니다. QMNC에 의해 생성됩니다.	아니오	예
QMNC	AQ 조정자 프로세스	메시지 큐를 모니터합니다. Qnnn 프로세스를 생성합니다.	아니오	예

자동 저장 영역 관리(ASM) 프로세스

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
ARBn	ASM 리밸런스 프로세스	ASM 파일 시스템에서 데이터 Extent를 리밸런스합니다. 가능한 프로세스는 ARB0 – ARB9 및 ARBA입니다.	아니오	아니오
ASMB	ASM 백그라운드 프로세스	자동 저장 영역 관리(ASM) Instance와 통신합니다.	아니오	아니오
DSKM	슬레이브 Diskmon 프로세스	RDBMS Instance 및 ASM Instance 사이의 통로 역할 및 I/O 한도 지정 정보, I/O Resource Manager 계획 및 SAGE 저장 영역에 대한 트랜잭션 커밋 캐시 정보를 통신하기 위한 마스터 Diskmon daemon 역 할을 수행합니다. SAGE 저장 영역이 사용되지 않는 경우, 슬레이브 diskmon 프로세스는 Instance 시작 후 자동으로 종료됩니다.	아니오	아니오
GMON	ASM 디스크 그룹 모니터 프로세스	ASM 디스크 그룹에서 디스크 멤버십을 유지 관리합니다.	아니오	아니오
MARK	재동기화 조정자용 AU 표시 프로세스	오프라인 디스크에 기록을 실패하면 ASM 할당 단위(AU: Allocation Unit)를 stale로 표시합니다.	아니오	아니오
RBAL	ASM 리밸런스 마스터 프로세스	자동 저장 영역 관리(ASM) Instance의 디스크 그룹에 대한 리밸런스 작업을 조정합니다. 자동 저장 영역 관리(ASM) 디스크에서 Global 열기를 수행합니다.	아니오	아니오
DRnn	ASM 디스크 재동기화 슬레이브 프로세스	오프라인 디스크의 컨텐트를 재동기화합니다. 디스크 온라인 SQL 명령이 오프라인인 디스크에서 실행되면 ASM이 DRnn을 생성합니다. 로드에 따라 두 개 이상의 슬레이브가 생성될 수 있습니다.	아니오	아니오

Data Guard 프로세스

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
DMON	Data Guard Broker 모니터 프로세스	Data Guard Broker가 시작될 때 시작됩니다. DMON은 로컬 데이터베이스 및 다른 데이터베이스의 DMON 프로세스와 상호 작용하여 요청된 기능을 수행하는 서버측 구성 요소입니다. 또한 Broker 구성에 대한 상태를 모니터하고 모든 데이터베이스의 구성 설명이 일관적인지 확인합니다.	아니오	아니오
FSFP	Data Guard Broker Fast Start Failover Pinger 프로세스	DGMGROL 클라이언트측 구성 요소에 통합된 관찰자 프로세스입니다. DG Broker 환경을 모니터하고 Primary에서 문제가 감지될 경우 Failover를 시작합니다.	아니오	아니오
INSV	Data Guard Broker Instance 슬레이브 프로세스	RAC 데이터베이스의 Instance 간에 Data Guard Broker 통신을 수행합니다.	아니오	아니오
LSP0	논리적 Standby 조정자 프로세스	다른 Data Guard Applier에 트랜잭션을 할당하고 트랜잭션 간 종속성이 유지되도록 조정합니다.	아니오	아니오
LSP1	논리적 Standby 딕셔너리 생성 프로세스	논리적 Standby를 위한 데이터 딕셔너리를 생성합니다.	아니오	아니오
LSP2	Logical Standby Set Guard 프로세스	논리적 Standby를 위한 Guard Standby 정보를 유지 관리합니다.	아니오	아니오
MRP0	관리되는 Standby Recovery 프로세스	RAC에서 XA Global 트랜잭션에 대한 투명한 지원을 제공합니다.	아니오	아니오
NSVn	Data Guard Broker NetSlave 프로세스	Data Guard 구성에서 데이터베이스 간의 Broker 네트워크 통신을 수행합니다. 가능한 프로세스는 NSV0-NSV9입니다.	아니오	아니오
RSM0	Data Guard Broker 작업자 프로세스	DMON 프로세스가 데이터베이스를 관리 및 모니터하는 데 사용됩니다.	아니오	아니오

RMAN 프로세스

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
CTWR	변경 사항 추적 기록자 프로세스	전체 데이터베이스를 나타내는 비트맵인 RMAN 변경 사항 추적 로그에 기록합니다. 비트맵에는 마지막 백업 시 SCN인 연관된 SCN이 포함됩니다.	아니오	아니오

Real Application Cluster 프로세스

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
ACMS	Atomic Control File to Memory Server 프로세스	Oracle RAC 환경에서 분산된 SGA 메모리 간성이 성공 시 global하게 생략되거나 Failure 시 global하게 중지되도록 보장합니다.	아니오	아니오
GTXn	Global 트랜잭션 프로세스	Oracle RAC 환경에서 XA Global 트랜잭션에 대한 투명한 지원을 제공합니다. 데이터베이스는 XA global 트랜잭션의 작업 로드에 따라 이러한 프로세스의 수를 자동으로 조정합니다. 가능한 프로세스는 GTX0–GTX9 및 GTx0–GTxj입니다.	아니오	아니오
LCK0	Instance enqueue 백그라운드 프로세스	Global enqueue 요청 및 Instance간 브로드캐스트를 관리합니다. 데이터 블록 이외의 리소스에 대한 모든 요청을 처리합니다.	아니오	아니오
LMD0	Global enqueue 서비스 Daemon 0 프로세스	Global enqueue 및 리소스에 대한 액세스를 제어 할 수 있도록 Global 캐시 서비스 enqueue에 대한 enqueue 관리자 서비스 요청을 관리합니다. LMD0 프로세스는 또한 Deadlock 감지 및 원격 enqueue 요청을 처리합니다. 원격 리소스 요청은 다른 Instance에서 시작된 요청입니다.	아니오	아니오

머리 글자어	프로세스 이름	설명	기본 작업에 필요	기본적으로 시작
LMON	Global enqueue 서비스 모니터 프로세스	전체 클러스터를 모니터하여 Global enqueue 및 리소스를 관리합니다. Instance를 관리하고 GCS(Global 캐시 서비스) 및 GES(Global enqueue 서비스)에 대한 Failure 및 관련 Recovery를 처리합니다. 특히 LMON은 Global 리소스와 연관된 Recovery 부분을 처리합니다. LMON 제공 서비스는 CGS(클러스터 그룹 서비스)로도 알려져 있습니다.	아니오	아니오
LMSn	Global 캐시 서비스 프로세스	원격 GCS(Global 캐시 서비스) 메시지를 처리합니다. LMS 프로세스 수는 클러스터의 노드 간에 전달되는 메시징 통신량에 따라 달라집니다.	아니오	아니오
PING	상호 연결 대기 시간 측정 프로세스	각 Instance 쌍에 대한 통신과 연관된 대기 시간을 평가합니다. 몇 초 간격으로 한 Instance(INSTANCE_NUMBER 값)의 프로세스가 두 개의 메시지를 각 Instance(TARGET_INSTANCE 값)에 전송합니다. 한 메시지의 크기는 500바이트이고 다른 메시지의 크기는 8KB입니다. 메시지는 대상 Instance의 PING 프로세스에 의해 수신되고 즉시 인정됩니다. 왕복 시간이 측정 및 수집됩니다.	아니오	아니오
RMSn	Oracle RAC 관리 프로세스	새 Instance가 클러스터에 추가될 때의 RAC 관련 리소스 생성과 같은 Oracle RAC에 대한 관리 효율성 작업을 수행합니다.	아니오	아니오
RSMN	원격 슬레이브 모니터 프로세스	원격 Instance에서 백그라운드 슬레이브 프로세스 생성 및 통신을 관리합니다.	아니오	아니오

부록 E
머리 글자어 및
용어

Oracle Internal & Oracle Academy
Use Only

단어	정의
Active Session Pool	Resource Group이나 Subplan에 허용된 현재 활성 세션 수입니다.
ADDM	Automatic Database Diagnostic Monitor
ASM	자동 저장 영역 관리(ASM)
ASM	Automatic Summary Management
ASMM	자동 공유 메모리 관리
ASSM	자동 세그먼트 공간 관리
ATO	자동 튜닝 옵티마이저
자동 PGA 메모리 관리	PGA 메모리 할당 방법을 단순화하고 향상시키는 오라클 데이터베이스 기능입니다.
자동 공유 메모리 관리	오라클 데이터베이스 instance에서 사용하는 가장 중요한 공유 메모리 구조 관리를 자동화하는 오라클 데이터베이스 기능입니다.
자동 저장 영역 관리(ASM)	오라클 데이터베이스 파일용으로 특별히 구축된 볼륨 관리자와 파일 시스템의 수직 통합을 제공하는 메커니즘입니다.
자동 데이터베이스 진단 관리	하향식 instance 분석을 수행하고, 문제와 가능한 원인을 식별하며, 문제 해결을 위한 권장 사항을 제공하는 유ти리티입니다.
자동 튜닝 옵티마이저	데이터베이스 내에서 다양한 SQL 성능 분석을 수행하는 데이터베이스 기능입니다.
자동 작업 로드 Repository	문제 감지와 자체 튜닝을 위해 성능 통계를 수집, 처리 및 유지 관리하는 infrastructure입니다.
보조 데이터베이스	복제 데이터베이스를 생성하거나 Tablespace Point-in-time Recovery를 수행할 때 사용되는 데이터베이스입니다.
AWR	자동 작업 로드 Repository
백업 피스(piece)	백업 셋에 포함되는 개별 파일입니다.
백업 셋	하나 이상의 데이터 또는 아카이브된 로그 파일의 복사본입니다. 빈 블록이 저장되지 않은 Image Copy와는 다릅니다.
블록 변경 사항 추적(Block change tracking)	변경 사항 추적 기록자(CTWR) 백그라운드 프로세스를 사용하여 모든 데이터베이스 변경 사항의 물리적 위치를 별도 파일에 기록하는 기능입니다.
블록 손상	데이터 블록이 손상된 경우입니다. 손상된 데이터 블록은 인식할 수 없는 Oracle 형식이거나 해당 내용이 내부적으로 일치하지 않는 블록입니다.
Block Media Recovery	Media Recovery의 최소 recovery 가능 단위를 데이터 파일에서 블록으로 낮추는 recovery 방식입니다.
BMR	Block Media Recovery
버퍼 캐시	데이터베이스에서 검색한 데이터 블록을 캐시하는 메모리 영역입니다.

단어	정의
CFS	Cluster File Storage
변경 사항 추적 파일	마지막 백업 이후에 변경한 데이터베이스 내용의 물리적 위치를 저장하는 데 사용되는 파일입니다.
채널	Target Database에 대한 링크 또는 연결입니다.
CLI	Command-line interpreter
클러스터	같은 데이터 블록을 공유하는 하나 이상의 테이블 그룹입니다.
CMAN	Oracle Connection Manager의 머리 글자어입니다. 넷 통신량 firewall 및 proxy 서버로 사용됩니다.
콘트롤 파일	모든 데이터 및 리두 로그 파일의 위치를 비롯하여 데이터베이스의 물리적 구조에 대한 정보를 포함하는 파일입니다.
CRS	Cluster Ready Services
데이터 블록	데이터베이스 내의 물리적 저장 영역의 최소 단위입니다. 데이터 블록은 데이터 행, 인덱스 정보 등을 포함합니다.
데이터 덕셔너리 캐시	메모리에서 덕셔너리 객체 정의를 보유하는 Shared Pool 내 메모리 영역입니다.
데이터 파일	데이터베이스의 데이터를 포함하는 파일입니다.
데이터베이스 Character Set 스캐너	오라클 데이터베이스를 새 데이터베이스 Character Set로 이전할 수 있는지 평가하는 유ти리티입니다.
DBA	데이터베이스 관리자
DBA	Data block address의 머리 글자어로 데이터베이스에서 데이터 블록을 고유하게 식별하는 데 사용되는 데이터 블록 주소입니다.
DBCA	Database Configuration Assistant
DBVERIFY	오프라인 데이터베이스에서 물리적 데이터 구조 무결성 검사를 수행하는 외부 명령행 유ти리티입니다.
DDL	Data Definition Language의 머리 글자어입니다. 데이터베이스 객체를 정의하고 조작하는 SQL 문의 한 유형입니다.
DML	Data Manipulation Language의 머리 글자어입니다. 데이터를 query하고 조작하는 SQL 문의 한 유형입니다.
EM	Enterprise Manager
emctl	Enterprise Manager Control의 머리 글자어로 Database Control, Oracle Agent 및 Oracle Management Server의 시작, 정지 및 상태 확인을 위한 유ти리티입니다.
인코딩된 Character Set	숫자 코드를 컴퓨터나 터미널이 표시하고 수신할 수 있는 문자로 매핑하는 Character Set입니다.
Enterprise Manager Database Control 콘솔	데이터베이스 관리에 사용되는 그래픽 인터페이스입니다.
EXTPROC	외부 코드 라이브러리

단어	정의
Fast recovery area	오라클 데이터베이스에서 모든 recovery 관련 파일과 작업을 위한 통합 저장 영역 위치입니다.
FGA	Fine-grained auditing
FGAC	Fine-Grained Access Control
Flash recovery area	이름이 Fast recovery area로 변경되었습니다.
Flashback 버퍼	Flashback Database 데이터를 저장하는 메모리 영역입니다.
Flashback Database	리두 데이터 대신 언두 데이터를 사용하여 데이터베이스를 recovery하는 새로운 recovery 방식입니다.
Flashback Drop	기존의 Point-in-time recovery에 의존하지 않고 DROP TABLE 문의 결과를 언두할 수 있는 기능입니다.
Flashback Table	테이블과 모든 종속 객체를 Recycle Bin에서 recovery 할 수 있는 명령입니다.
Flashback Transaction Query	트랜잭션 레벨에서 데이터베이스에 대한 변경 사항을 볼 수 있는 진단 도구입니다.
Flashback Versions Query	변경한 트랜잭션의 해당 식별자와 함께 행의 변경 기록을 제공하는 query 구문입니다.
형식 마스크 요소	문자열에 저장된 일시 또는 숫자 데이터 형식을 설명하는 문자 리터럴입니다.
증가 추세 보고서	데이터베이스 세그먼트 증가에 대한 분석입니다.
Globalization Support	유틸리티와 오류 메시지, 정렬 순서, 알파벳, 달력, 날짜, 시간, 통화 및 숫자가 자국어에 자동으로 적용되도록 보장하는 기능입니다.
이미지 복사본	비트별로 동일한 데이터베이스 파일의 복사본입니다.
Incarnation	물리적 데이터베이스의 개별 버전입니다. 데이터베이스 Incarnation은 RESETLOGS 옵션으로 열 때 변경되지만 필요한 리두를 사용할 수 있으면 이전 Incarnation에서 백업을 recovery 할 수 있습니다.
인덱스 구성 테이블 (Index-organized table)	테이블 모양이지만 B*Tree 구조로 데이터를 저장하는 데이터베이스 구조입니다.
init.ora 또는 init<sid>.ora	시작 시 데이터베이스 instance의 구성 및 실행 방식을 제어하는 초기화 파라미터 파일입니다. "파라미터 파일"로도 알려져 있습니다.
instance	오라클 데이터베이스에 액세스하는 데 사용되는 공유 메모리와 프로세스 모음입니다.
IPC	Internal Process Communication
isqlplusctl	iSQL*Plus 리스너 프로세스를 시작 및 중지하는 Control 유ти리티입니다.
ISV	Independent software vendor

단어	정의
Java pool	JVM(Java Virtual Machine) 내의 모든 세션 전용 Java 코드와 데이터에 사용되는 SGA의 메모리 영역입니다.
JDBC	Java Database Connectivity
jnnn	작업 큐 프로세스로, 일정이 잡힌 작업을 실행합니다.
Keep 버퍼 캐시	오랫동안 버퍼 캐시에 데이터를 캐시하는 데 사용되는 SGA의 메모리 영역입니다.
언어 및 Character Set 파일 스캐너	알 수 없는 파일 텍스트의 언어와 Character Set를 결정하는 통계 기반 유ти리티입니다.
Large pool	대규모 I/O 요청을 버퍼링하는 데 사용되는 메모리 저장 영역(선택 사항)입니다.
LCSSCAN	언어 및 Character Set 파일 스캐너
LEGATO® NetWorker, 단일 서버 버전	Oracle Database 10g에 포함된 소프트웨어로 Recovery Manager 유ти리티를 사용하여 테이프 드라이브에 기록합니다.
라이브러리 캐시	PL/SQL 블록 및 SQL 문의 완전히 구분 분석되거나 컴파일된 표현을 포함하는 Shared Pool 내의 메모리 영역입니다.
언어 정렬	문자 인코딩 방식의 해당 숫자 값이 아닌 문자의 알파벳 시퀀스와 일치하는 정렬 시퀀스를 생성하는 기능입니다.
리스너	로컬이 아닌 모든 유저 연결을 위한 Oracle instance에 대한 게이트웨이입니다.
로케일	특정 지역의 언어 및 문화 환경 설정에 대한 정보 모음입니다.
로케일 변형	언어 종속적인 영역 정의입니다.
LSSV	LEGATO® NetWorker, 단일 서버 버전
Media management library	RMAN이 테이프에 쓰거나 테이프에서 읽을 때 사용하는 인터페이스입니다.
Memory Advisor	메모리 구조 크기를 튜닝할 수 있는 Enterprise Manager 기능입니다.
MMAN(메모리 관리자)	SGA 메모리 Broker 역할을 하고 메모리 구성 요소 크기를 조정하는 데이터베이스 백그라운드 프로세스입니다.
Metric	일부 데이터베이스 또는 instance 특성을 측정하는 것입니다.
MML	Media management library
MMON	Management Monitor Process의 머리 글자입니다. 이 프로세스는 metric이 임계값을 위반할 때마다 alert를 발행합니다. 최근에 수정된 SQL 객체에 대한 통계를 캡처합니다.
국가별 언어 지원	데이터베이스 클라이언트와 데이터베이스 서버의 로케일별 동작을 결정하는 파라미터 및 파일입니다.

단어	정의
nK 블록 크기 버퍼	기본 데이터베이스 블록 크기와 크기가 다른 데이터 블록을 캐시하는 SGA의 메모리 영역으로 Transportable 데이터스페이스를 지원하는 데 사용됩니다.
NLS	국가별 언어 지원
NLS Runtime Library	적절한 텍스트와 문자 처리 및 언어 규칙 조작을 가능하게 하는 포괄적인 언어 독립적인 기능 모음입니다.
NLS_LANG	데이터베이스에서 사용하는 언어, 지역 및 Character Set를 지정하는 데 필요한 환경 변수입니다.
NLSRTL	National Language Support Runtime Library
NMP	명명된 파이프
OC4J	Oracle Application Server Containers for J2EE
OMF	Oracle Managed Files
옵티마이저 통계	데이터베이스와 데이터베이스의 객체를 설명하는 통계이며 Query 옵티마이저가 각 SQL 문에 가장 적합한 실행 계획을 선택할 때 사용합니다.
OUI	Oracle Universal Installer
Oracle Locale Builder	로케일별 데이터를 쉽게 보고 수정하며 정의할 수 있는 Graphic User Interface를 제공하는 기능입니다.
Oracle Managed Files	전용 디스크 영역 내에서 오라클 데이터베이스 파일의 생성, 이름 지정 및 삭제를 관리하는 오라클 데이터베이스 기능입니다.
Oracle Net	Oracle Database 10g와 클라이언트 또는 middle-tier 응용 프로그램 간의 네트워크 연결을 가능하게 하는 인터페이스입니다.
Oracle Shared Server	많은 User Process에서 작은 수의 서버 프로세스를 공유하여 서버 프로세스 수를 최소화하고 사용 가능한 시스템 리소스 사용을 최대화할 수 있는 데이터베이스 서버 구성입니다.
ORACLE_BASE	OFA 구조의 기반을 가리키는 데 사용되는 환경 변수입니다.
ORACLE_HOME	Oracle 소프트웨어가 들어 있는 디렉토리를 식별하는 데 사용되는 환경 변수입니다.
ORACLE_SID	기본 데이터베이스 instance 이름을 지정하는 데 사용되는 환경 변수입니다.
패키지	논리적으로 연관된 프로시저 및 함수 정의 모음입니다. 이러한 프로시저와 함수는 Package Body에서 구현합니다.
병렬화	RMAN 백업과 recovery 작업에 다중 채널을 할당합니다.
PGA	Program Global Area
PGA Advisor	작업 영역에 대한 자세한 통계를 제공하고 작업 로드 특성에 따른 PGA(Program Global Area) 메모리의 최적 사용에 대한 권장 사항을 제공하는 Enterprise Manager 기능입니다.

단어	정의
파이프	하나의 프로세스가 다른 프로세스로 정보를 전달하기 위해 사용하는 메모리 영역입니다.
전용(private) SQL	바인드 정보 및 런타임 메모리 구조와 같은 데이터가 들어 있는 PGA의 메모리 영역입니다.
권한	특정 유형의 SQL 문을 실행할 수 있는 권한입니다. 권한은 객체 및 시스템이라는 두 가지 기본 형태가 있습니다.
테이블스페이스 사전 모니터링(Proactive Monitoring)	테이블스페이스 디스크 공간 사용을 관리하는 Oracle Database 10g 기능입니다.
Program Global Area	프로세스가 사용하는 전용(private) 메모리 영역입니다.
Recovery catalog	백업 작업과 관련된 기록 데이터를 보관하는 별도의 데이터베이스입니다.
Recovery Manager	데이터베이스 파일 백업 및 복원에 사용되는 Oracle 유ти리티입니다.
Recycle bin	삭제된 객체의 원래 이름과 시스템 생성 이름 간의 관계를 유지 관리하는 데이터 딕셔너리 테이블입니다.
Recycle 버퍼 캐시	버퍼 캐시에서 빠르게 삭제되는 데이터를 보유하는 SGA의 메모리 영역입니다.
리두 로그 버퍼	디스크에 쓸 수 있을 때까지 리두 정보를 캐시하는 메모리 영역입니다.
Redo Log File Sizing Advisor	리두 로그 파일 크기 권고를 제공하는 Enterprise Manager 기능입니다.
Resource Manager	오라클 데이터베이스 서버에 더 강력한 리소스 관리 결정 권한을 제공하여 비효율적인 운영 체제 관리로 인해 발생하는 문제를 방지하는 오라클 데이터베이스 기능입니다.
Resumable space allocation	공간 할당이 실패한 경우 대규모 데이터베이스 작업 실행을 일시 중지했다가 나중에 재시작할 수 있는 방법입니다.
RMAN	Recovery Manager
RMAN Repository	데이터베이스 백업 및 recovery 작업에 대한 메타 데이터를 유지 관리하는 저장 영역 구조입니다.
스케줄러	데이터베이스 관리자와 응용 프로그램 개발자가 데이터베이스 환경에서 여러 작업을 수행하는 시기와 위치를 제어할 수 있도록 하는 새로운 데이터베이스 기능입니다.
SCN	시스템 변경 번호
Segment Advisor	객체 공간 문제를 모니터하고 증가 추세를 분석하는 Advisor입니다.

단어	정의
Segment Resource Estimator	새 세그먼트 리소스 예측 기능을 사용하여 새 세그먼트를 생성해야 하는 리소스 양을 예측할 수 있습니다.
서버 세션	클라이언트 유ти리티가 Target Database에 연결하기 위해 호출하는 서버 프로세스(UNIX) 또는 스레드(Windows NT/2000)입니다.
세션 메모리	세션 변수 및 세션과 관련된 기타 정보를 보유하기 위해 할당되는 PGA의 메모리입니다.
SGA	System Global Area의 머리 글자어입니다. 모든 서버 및 백그라운드 프로세스가 공유하는 메모리 영역입니다.
SGA Advisor	SGA 관련 파라미터 설정에 대한 권장 사항을 제공하는 Advisor입니다.
Shared pool	유저 간에 공유할 수 있는 여러 구성을 캐시하는 메모리 영역입니다.
Shrink Advisor	Segment Advisor를 참조하십시오.
SID	시스템 식별자입니다. 기본값은 데이터베이스 이름이며 지정된 서버의 instance를 고유하게 식별합니다.
SQL	구조적 질의어(SQL)
SQL Access Advisor	최적의 데이터 액세스 경로를 파악하는 도구입니다(예: 인덱스 및 materialized view 사용).
SQL Tuning Advisor	SQL 문에 대한 튜닝 조언을 제공하는 도구입니다.
Statspack	성능 데이터의 수집, 자동화, 저장 및 보기할 수 있는 SQL, PL/SQL 및 SQL*Plus 스크립트 집합입니다. 이 기능은 Automatic Workload Repository로 대체되었습니다.
Streams pool	Oracle Streams에서 사용하는 SGA의 메모리 영역(선택 사항)입니다.
시스템 통계	Query 옵티마이저에 시스템의 하드웨어 특성(예: I/O 및 CPU 성능 및 활용)을 설명하는 통계입니다.
테이블스페이스	데이터 파일의 논리적 그룹입니다.
Target Database	연결하려는 데이터베이스입니다.
임계값	metric 값이 비교되는 경계 값입니다.
Undo Advisor	지정된 시간 동안 flashback을 지원하는 데 필요한 파라미터 값과 추가 공간 크기를 제안하는 Enterprise Manager 기능입니다.
언두 데이터	DML 트랜잭션이 데이터를 변경할 때마다 저장되는 원래 데이터의 복사본입니다. 언두 데이터는 트랜잭션을 롤백하는 데 사용되며 변경되는 데이터에 대해 읽기 일관성이 있는 보기를 제공합니다.
User Global Area	Oracle Shared Server 세션에 대한 세션 정보를 포함하는 Shared Pool 또는 Large Pool 내의 메모리 영역입니다.

단어	정의
UTC	Universal Time Coordinates 의 머리 글자어입니다. Uniform Audit Trail 의 세계 시간 기록입니다.
VPD	Virtual Private Database
대기 이벤트	서버 프로세스나 스레드에 의해 증가되는 통계로 이벤트가 완료될 때까지 대기해야 프로세스를 계속할 수 있음을 나타냅니다.
작업 영역	메모리가 많이 사용되는 정렬, hash join 및 기타 작업에 사용하기 위해 PGA에서 할당된 전용(private) 메모리입니다.
작업 로드 Repository	AWR을 참조하십시오.

Oracle Internal & Oracle Academy
Use Only



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

목표

이 단원을 마치면 Oracle Restart를 사용하여 구성 요소를 관리할 수 있습니다.



Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

Oracle Restart

Oracle Restart는 독립형 오라클 데이터베이스에 대한 고가용성 솔루션을 구현합니다.

- 다음 구성 요소를 모니터 및 재시작할 수 있음:
 - 데이터베이스 Instance
 - Oracle Net 리스너
 - 데이터베이스 서비스
 - 자동 저장 영역 관리(ASM) Instance
 - ASM 디스크 그룹
 - ONS/eONS(Oracle Notification Services)
- 주기적인 검사 작업을 실행하여 구성 요소의 상태를 모니터합니다.
- 오라클 데이터베이스 홈과 별개로 설치하는 Oracle 그리드 Infrastructure 홈에서 실행됩니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Restart

Oracle Restart는 오라클 데이터베이스의 가용성을 향상시킬 수 있도록 설계되었습니다. Oracle Restart는 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
 - ONS/eONS(Oracle Notification Services): Failover 시 FAN(Fast Application Notification) 이벤트를 통합 클라이언트에 전송하는 서비스입니다. eONS는 Oracle Enterprise Manager가 Oracle Restart에 의해 관리되는 구성 요소의 상태에 대한 변경 알림을 받기 위해 사용됩니다.
- ASM 디스크 그룹을 재시작하면 디스크 그룹이 마운트됩니다. ONS 재시작 기능은 FAN을 통한 기본 데이터베이스 및 standby database 사이의 자동 Failover를 위해 Oracle Data Guard 설치에서만 적용할 수 있습니다.

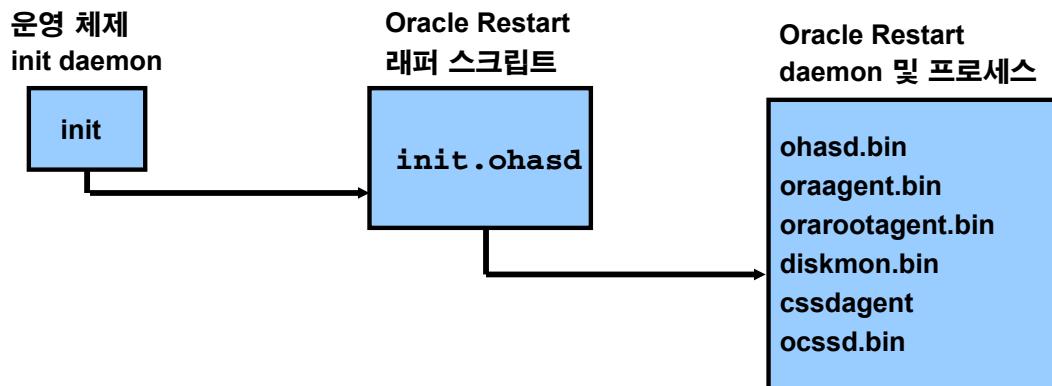
Oracle Restart(계속)

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.

Oracle Internal & Oracle Academy
Use Only

Oracle Restart 프로세스 시작

- Oracle Restart는 OS init daemon에 의해 시작됩니다.



- Oracle Restart를 설치하면 시스템을 시작할 때마다 Oracle Restart가 시작되도록 /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.

Oracle Restart 프로세스 시작

Oracle Restart를 설치하면 래퍼 스크립트를 시작하기 위한 항목들이 /etc/inittab 운영 체제 파일에 배치됩니다. 래퍼 스크립트는 환경 변수를 설정한 다음 Oracle Restart daemon 및 프로세스를 시작합니다.

명령을 사용하여 Oracle Restart를 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. 또한 daemon이 실패한 경우 재생성된 작업으로 인해 init 프로세스에서 daemon이 재시작됩니다.

일부 Oracle Restart 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.

Oracle Restart 제어

CRSCTL 유ти리티를 사용하여 Oracle Restart의 상태를 제어할 수 있습니다.

- Oracle Restart 구성을 표시하려면:

```
$ crsctl config has
```

- Oracle Restart의 자동 재시작을 활성화 또는 비활성화하려면:

```
$ crsctl [ enable | disable ] has
```

- Oracle Restart를 시작 또는 정지하려면:

```
$ crsctl [ start | stop ] has
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Restart 제어

CRSCTL 유ти리티를 사용하여 Oracle Restart의 상태를 제어할 수 있습니다. 이 유ти리티를 사용하면 다음과 같이 자동 시작 기능을 활성화하거나 비활성화할지를 결정할 수 있습니다.

```
$ crsctl config has  
CRS-4622: Oracle High Availability Services autostart is  
enabled.
```

Oracle 그리드 Infrastructure를 설치하는 동안 다음과 같이 래퍼 스크립트를 시작하기 위한 항목들이 /etc/inittab 운영 체제 파일에 배치됩니다.

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

래퍼 스크립트는 환경 변수를 설정한 후 다른 관련 프로세스와 함께 Oracle High Availability Services Daemon(ohas)을 시작합니다. 래퍼 스크립트는 root 유저로 실행됩니다.

CRSCTL 유ти리티를 사용하여 Oracle 그리드 Infrastructure 자동 재시작을 비활성화할 경우 /etc/inittab 파일의 항목이 제거되지 않습니다. ohas에 대한 콘트롤 파일은 ohas 상태를 제어하고 자동 재시작이 활성화 또는 비활성화되었는지를 확인하기 위해 사용됩니다. 이러한 콘트롤 파일은 SCLS_SCR 파일로 알려져 있습니다. Linux의 경우 콘트롤 파일의 위치는 다음과 같이 정의됩니다.

```
/etc/oracle/scls_scr/$HOST/<Oracle Restart owner> and  
/etc/oracle/scls_scr/$HOST/root
```

Oracle Restart 제어(계속)

CRSCTL 유ти리티를 사용하여 Oracle Restart를 정지할 경우 Oracle Restart가 현재 관리하는 모든 구성 요소도 정지됩니다.

```
$ crsctl stop has
CRS-4549: Stopping resources.
CRS-2673: Attempting to stop 'ora.diskmon' on 'host01'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host01'
CRS-2677: Stop of 'ora.DATA.dg' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2675: Stop of 'ora.diskmon' on 'host01' succeeded
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host01' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'host01'
CRS-2677: Stop of 'ora.diskmon' on 'host01' succeeded
CRS-4133: Oracle High Availability Services has been stopped.
```

위 예제에서는 오라클 데이터베이스 소프트웨어가 설치되지 않았습니다. 오라클 데이터베이스가 Oracle Restart에 등록되었거나 ASM을 사용 중인 경우 오라클 데이터베이스도 정지됩니다.

CRSCTL 유ти리티로 Oracle Restart를 시작할 경우 시작된 각 구성 요소가 표준 출력에 표시되지 않습니다.

```
$ crsctl start has
CRS-4123: Oracle High Availability Services has been started.
```

주: 래퍼 스크립트를 직접 호출하여 Oracle 그리드 Infrastructure 프로세스를 시작할 수 없습니다.

올바른 SRVCTL 유ти리티 선택

- **ASM Instance, 디스크 그룹, 리스너 및 ONS를 사용할 때 Oracle 그리드 Infrastructure 홈에서 Server Control (SRVCTL) 유ти리티를 호출합니다.**

```
$ export ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid  
$ $ORACLE_HOME/bin/srvctl command component options
```

- **데이터베이스 또는 데이터베이스 Instance를 사용할 때 오라클 데이터베이스 홈에서 SRVCTL 유ти리티를 호출합니다.**

```
$ export  
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1  
$ $ORACLE_HOME/bin/srvctl command component options
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

올바른 SRVCTL 유ти리티 선택

Oracle Restart에는 Oracle Restart 구성 요소를 시작, 정지 및 관리하는 데 사용되는 SRVCTL 유ти리티가 포함됩니다. Oracle 그리드 Infrastructure 소프트웨어 외에도 오라클 데이터베이스 소프트웨어가 설치된 경우 각 ORACLE_HOME 위치에 SRVCTL 유ти리티의 복사본이 있습니다. SRVCTL 유ти리티를 실행할 올바른 ORACLE_HOME 위치를 확인해야 합니다. ASM Instance, ASM 디스크 그룹, Oracle Net 리스너 및 ONS를 관리할 때는 그리드 Infrastructure 소프트웨어 홈 디렉토리에서 SRVCTL 유ти리티를 실행해야 합니다. 오라클 데이터베이스 Instance를 관리할 때는 오라클 데이터베이스 소프트웨어 홈 디렉토리에서 SRVCTL 유ти리티를 실행해야 합니다. SRVCTL 유ти리티의 현재 매팅 위치를 확인하려면 다음과 같이 which 명령을 사용합니다.

```
$ which srvctl  
/u01/app/oracle/product/11.2.0/grid/bin/srvctl
```

주: Oracle Net 리스너의 경우 오라클 데이터베이스 소프트웨어 이전에 Oracle 그리드 Infrastructure 가 설치된 것으로 가정합니다. Oracle Restart가 기준 오라클 데이터베이스 설치에 추가된 경우 Oracle Net 리스너를 오라클 데이터베이스 홈 디렉토리에서 실행할 수 있습니다. 이 경우 오라클 데이터베이스 홈에서 SRVCTL 유ти리티를 사용하여 Oracle Net 리스너를 관리합니다.

Oracle Restart 구성

오라클 유ти리티는 Oracle Restart 구성을 자동으로 갱신합니다.

작업 및 Oracle Restart 구성 생성	구성에 자동으로 추가됩니까?
OUI 또는 DBCA를 사용하여 데이터베이스 생성	예
SQL 문으로 데이터베이스 생성	아니오
OUI, DBCA 또는 ASMCA로 ASM Instance 생성	예
디스크 그룹 생성(모든 방법)	예
NETCA로 리스너 추가	예
SRVCTL로 데이터베이스 서비스 생성	예
SERVICE_NAMES 초기화 파라미터를 수정하여 데이터베이스 서비스 생성	아니오
DBMS_SERVICE.CREATE_SERVICE로 데이터베이스 서비스 생성	아니오

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Restart 구성

Oracle Restart는 Oracle Restart가 관리하는 모든 구성 요소 리스트를 유지 관리하고 각 구성 요소에 대한 구성 정보를 유지 관리합니다. 이러한 모든 정보는 총체적으로 "Oracle Restart 구성"이라고 부릅니다. Oracle Restart가 설치되면 오라클 유ти리티를 사용하여 오라클 구성 요소를 생성하는 많은 작업들에서 구성 요소가 Oracle Restart 구성에 자동으로 추가됩니다. 구성 요소가 오라클 유ти리티를 사용하지 않고 수동으로 생성된 경우 필요에 따라 SRVCTL 명령을 사용하여 해당 구성 요소를 Oracle Restart 구성에 추가할 수 있습니다. 위 슬라이드에 표시된 표에서는 Oracle Restart 구성에 구성 요소를 자동으로 생성하는 작업과 Oracle Restart 구성을 갱신하지 않는 작업을 생성하는 작업이 표시됩니다.

주: 작업을 삭제할 때도 동일한 원칙이 적용됩니다.

SRVCTL 유ти리티 사용

- SRVCTL 유ти리티는 다음 구문과 함께 Oracle Restart 구성 요소를 시작, 정지 및 관리하는 데 사용됩니다.

```
$ srvctl command component options
```

- 다음 명령 및 구성 요소가 지원됩니다.

명령	add config disable enable getenv modify remove setenv start status stop unsetenv
구성 요소	asm db dg filesystem home lsnr serv ons eons

ORACLE

Copyright © 2010, Oracle. All rights reserved.

SRVCTL 유ти리티 사용

Oracle Restart를 사용 중인 경우 오라클은 SRVCTL 유ти리티를 사용하여 모든 Oracle Restart 구성 요소를 시작, 정지 및 관리할 것을 적극 권장합니다. SRVCTL 유ти리티가 권장되는 이유는 다음과 같습니다.

- 구성 요소 간의 모든 종속성이 유지 관리됩니다. 따라서 Oracle Restart가 모든 종속된 구성 요소를 먼저 시작하거나 중지할 수 있습니다.
- 구성 요소가 해당 Oracle Restart 구성에 따라 시작됩니다.
- 구성 요소에 대해 Oracle Restart 구성에 저장된 환경 변수가 설정됩니다.

Oracle Restart 구성 요소는 리스너 콘트롤(LSNRCTL) 유ти리티 또는 SQL*Plus와 같은 다른 유ти리티로 시작할 수 있지만 위에 나열된 이점은 다른 유ти리티에서는 제공되지 않습니다. Oracle Restart의 SRVCTL 유ти리티는 12개의 명령과 9개의 구성 요소를 지원합니다. 허용되는 옵션은 각 명령 및 구성 요소 조합에 따라 다릅니다. SRVCTL 유ти리티 구문은 다음과 같습니다.

```
srvctl command component options
```

여기서

- command는 start, stop 또는 remove와 같은 동사입니다.
- component는 SRVCTL이 명령을 수행하는 객체(예: 데이터베이스)입니다.
- options는 추가 파라미터를 포함하도록 이전 명령의 사용 범위를 확장합니다.

SRVCTL 유ти리티에 대한 도움말 보기

SRVCTL 유ти리티는 해당 명령, 구성 요소 및 옵션에 대한 온라인 도움말을 제공합니다.

- 일반 사용에 대한 도움말을 보려면 다음을 수행하십시오.

```
$ srvctl -h
```

- 특정 명령에 대한 도움말을 보려면 다음을 수행하십시오.

```
$ srvctl command -h
```

- 특정 명령 및 구성 요소에 대한 도움말을 보려면 다음을 수행하십시오.

```
$ srvctl command component -h
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

SRVCTL 유ти리티에 대한 도움말 보기

SRVCTL 유ти리티는 해당 명령, 구성 요소 및 옵션에 대한 자세한 온라인 도움말을 제공합니다. 온라인 도움말을 표시하려면 도움말(-h) 옵션을 사용하여 사용 정보를 표시합니다. 도움말(-h) 옵션이 지정된 유일한 파라미터인 경우 SRVCTL은 각 명령 및 구성 요소 조합에 대해 사용되는 가장 일반적인 옵션들을 포함하는 모든 명령의 일반 outline을 표시합니다. 이 outline은 모든 지원되는 옵션에 대한 전체 리스트가 될 수 없습니다. 보다 자세하고 완전한 정보를 보려면 특정 명령 또는 특정 명령과 구성 요소의 조합에 대해 도움말(-h) 옵션을 사용할 수 있습니다.

SRVCTL 유ти리티를 사용하여 구성 요소 시작

오라클은 SRVCTL 유ти리티를 사용하여 모든 구성 요소를 시작할 것을 권장합니다.

- 개별 구성 요소 시작에 대한 예제는 다음과 같습니다.

```
$ srvctl start database -d PROD -o mount  
$ srvctl start listener -l crmlistener  
$ srvctl start service -d PROD -s "service1,service2"  
$ srvctl start diskgroup -g "DATA,FRA"  
$ srvctl start asm  
$ srvctl start eons -v  
$ srvctl start ons
```

- 지정된 Oracle 흄에서 모든 Oracle Restart 구성 요소 시작에 대한 예제:

```
$ srvctl start home -o oracle_home -s state_file
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

SRVCTL 유ти리티를 사용하여 구성 요소 시작

SRVCTL 유ти리티는 필요한 종속 구성 요소와 함께 개별 구성 요소를 시작하는 데 사용할 수 있습니다. 예를 들어, `srvctl start database -d PROD` 명령은 해당 구성 요소가 Oracle Restart에 의해 관리되도록 정의되고 PROD 데이터베이스에 대해 종속 구성 요소로 나열된 경우 리스너, ASM Instance 및 여러 디스크 그룹을 시작할 수도 있습니다.

SRVCTL 유ти리티는 또한 지정된 Oracle 흄과 연관되고 다음 명령에 의해 Oracle Restart에 대해 구성된 모든 구성 요소를 시작하는 데 사용할 수도 있습니다.

```
srvctl start home -o /u01/app/oracle/product/11.2.0/dbhome_1  
-s /usr/local/bin/group_state_file
```

state 파일은 Oracle 흄에 있는 구성 요소에 대한 현재 상태 정보를 포함하며 `srvctl status home` 명령이 실행될 때 생성됩니다. 이 파일은 state 파일 옵션(-s)으로 표시되며 state 파일의 전체 경로를 지정해야 합니다. state 파일은 모든 디렉토리에 생성할 수 있습니다.

주: 슬라이드에 표시된 옵션은 가장 일반적인 옵션을 나타내며 전체 리스트가 아닙니다. 도움말 옵션(-h)을 사용하여 각 명령에 대한 모든 사용 가능한 전체 리스트를 볼 수 있습니다.

SRVCTL 유ти리티를 사용하여 구성 요소 정지

오라클은 SRVCTL 유ти리티를 사용하여 모든 구성 요소를 정지할 것을 권장합니다.

- 개별 구성 요소 정지에 대한 예제는 다음과 같습니다.

```
$ srvctl stop database -d PROD -o transactional  
$ srvctl stop listener -l crmlistener -f  
$ srvctl stop service -d PROD -s "service1,service2"  
$ srvctl stop diskgroup -g "DATA,FRA" -f  
$ srvctl stop asm -o immediate -f  
$ srvctl stop eons -v  
$ srvctl stop ons
```

- 지정된 Oracle 홈에서 모든 Oracle Restart 구성 요소 정지에 대한 예제:

```
$ srvctl stop home -o oracle_home -s state_file -f
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

SRVCTL 유ти리티를 사용하여 구성 요소 정지

SRVCTL 유ти리티는 정지해야 하는 종속 구성 요소와 함께 개별 구성 요소를 정지하는 데 사용할 수 있습니다. 예를 들어, `srvctl stop diskgroup -g "DATA" -f` 명령은 파일이 열려 있더라도 디스크 그룹을 강제로 마운트 해제합니다. 또한 DATA 디스크 그룹을 사용하는 모든 데이터베이스 Instance를 정지합니다.

SRVCTL 유ти리티는 또한 지정된 Oracle 홈과 연관되고 다음 명령에 의해 Oracle Restart에 대해 구성된 모든 구성 요소를 정지하는 데 사용할 수도 있습니다.

```
srvctl stop home -o /u01/app/oracle/product/11.2.0/dbhome_1 -  
s /usr/local/bin/group_state_file -f
```

이 기능은 소프트웨어 binary file에 패치를 적용해야 할 때와 같이 모든 구성 요소를 정지해야 할 경우에 매우 유용합니다.

주: 슬라이드에 표시된 옵션은 가장 일반적인 옵션을 나타내며 전체 리스트가 아닙니다. 도움말 옵션(-h)을 사용하여 각 명령에 대한 모든 사용 가능한 전체 리스트를 볼 수 있습니다.

구성 요소 상태 보기

- **status 명령을 사용하여 Oracle Restart에서 관리되는 모든 구성 요소에 대한 실행 상태를 볼 수 있습니다.**
- **데이터베이스에 대한 실행 상태 표시:**

```
$ srvctl status database -d orcl  
Database is running.
```

- **리스너 상태 표시:**

```
$ srvctl status lsnr  
Listener LISTENER is enabled  
Listener LISTENER is running on node(s): host01
```

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

구성 요소 상태 보기

SRVCTL 유ти리티를 사용하여 Oracle Restart에서 관리되는 모든 구성 요소에 대한 실행 상태 (실행 중 또는 실행 중이 아님)를 볼 수 있습니다. 일부 구성 요소의 경우 추가 정보가 표시됩니다. 명령의 형식은 다음과 같습니다.

```
srvctl status object [options]
```

여기서 *object*는 다음 중 하나일 수 있습니다.

- *asm*: ASM instance
- *db*: 데이터베이스 instance
- *dg*: Oracle ASM 디스크 그룹
- *filesystem*: Oracle ASM 파일 시스템
- *home*: Oracle 흄 또는 Oracle Clusterware 흄
- *lsnr*: Oracle Net 리스너
- *ons*, *eons*: Oracle Notification Services
- *serv*: 데이터베이스 서비스

위에 표시된 각 객체의 옵션 리스트를 보려면 *Oracle Database Administrators Guide*를 참조하십시오.

구성 요소에 대한 Oracle Restart 구성 표시

- config 명령을 사용하여 구성 요소의 Oracle Restart 구성을 표시할 수 있습니다.
- 데이터베이스에 대한 Oracle Restart 구성 표시:

```
$ srvctl config database -d orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain: example.com
Start options: open
Stop options: immediate
Database role:
Management policy: automatic
Disk Groups: DATA,FRA
Services: east,sales
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

구성 요소에 대한 Oracle Restart 구성 표시

SRVCTL 유ти리티에서는 config 명령을 사용하여 구성 요소의 Oracle Restart 구성을 표시할 수 있습니다. config 명령은 database, service, asm, listener, ons 및 eons 구성 요소에 대해 사용할 수 있습니다.

Oracle Restart 구성 요소의 구성은 SRVCTL 유ти리티의 modify 명령을 사용하여 수정할 수 있습니다. 다음 구문에서는 서버 파라미터 파일(SPFILE)에 대해 서로 다른 비표준 디렉토리를 사용하기 위해 orcl의 고유한 이름으로 데이터베이스를 수정하는 방법 예제를 보여 줍니다.

```
srvctl modify database -d orcl -p
/usr/local/oracle/spfileorcl.ora
```

Oracle Restart 구성에 구성 요소를 수동으로 추가

구성 요소는 add 명령을 사용하여 Oracle Restart 구성에 수동으로 추가할 수 있습니다.

- NETCA로 생성되지 않은 새 리스너를 정의하려면 다음 구문을 사용하십시오.

```
$ srvctl add listener -l MYLISTENER -p TCP:1525 -o /u01/app/oracle/product/11.2.0/grid
```

- 새 리스너의 네트워킹 파일에 대한 비기본 위치를 지정하려면 다음 구문을 사용하십시오.

```
$ srvctl setenv listener -l MYLISTENER -t TNS_ADMIN=/usr/local/oracle
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Restart 구성에 구성 요소를 수동으로 추가

SRVCTL 유ти리티의 add 명령을 사용하면 Oracle Restart 구성에 구성 요소를 수동으로 추가할 수 있습니다. 구성 요소가 NETCA, DBCA, ASMCA 또는 OUI와 같은 오라클 유ти리티로 생성된 경우 Oracle Restart 구성에 자동으로 추가되므로 수동으로 추가할 필요가 없습니다.

슬라이드에서는 MYLISTENER라는 새 리스너를 Oracle Restart 구성에 수동으로 추가하는 방법을 보여 줍니다. 리스너는 Grid 홈 설치의 소프트웨어 binary file을 사용하지만 네트워킹 파일에 대해 비표준 위치에 따라 다릅니다. 이 예제에서는 listener.ora 네트워킹 파일이 /usr/local/oracle 디렉토리에 생성되었다고 가정합니다. srvctl 유ти리티의 setenv 명령은 특정 구성 요소에 필요할 수 있는 환경 변수를 정의하기 위해 사용됩니다. TNS_ADMIN 환경 변수는 listener.ora 파일의 비기본 위치로 설정되었으며, MYLISTENER라는 리스너에 대해서만 정의되어 있습니다. 이러한 설정은 이미 존재하고 네트워킹 파일에 대해 다른 디렉토리를 사용하는 다른 리스너에 영향을 주지 않습니다.

SRVCTL 유ти리티에는 Oracle Restart 구성에서 구성 요소를 수동으로 삭제하기 위한 remove 명령이 포함됩니다. 위에서 생성된 리스너를 삭제하려면 다음 구문을 사용합니다.

```
srvctl remove listener -l mylistener -f
```

그리면 리스너와 연관된 환경 변수도 제거됩니다.

퀴즈

다음 중 어느 항목을 사용할 때 Oracle 그리드 Infrastructure에서 SRVCTL 유ти리티를 호출합니까?

- 1. 리스너**
- 2. ASM instance**
- 3. 데이터베이스 instance**
- 4. ASM 디스크 그룹**

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

해답: 1, 2 및 4

요약

이 단원에서는 Oracle Restart를 사용하여 구성 요소를 관리하는 방법을 배웠습니다.



Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

연습 3-1: 개요

이 연습에서는 Oracle Restart를 사용하여 구성 요소를 관리합니다.



Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only



지속적인 교육 및 추가 정보

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

후속 학습

"기술 업계에서 경쟁력을 유지하려면 끊임없이 배워야 합니다.
항상 더 효율적으로 일을 처리할 수 있는 방법과 새 기술을
찾으십시오. 우리 업계에서 발전하지 않는 자에게 보상이란
없습니다".

–John Hall, Oracle University 부사장

이 부록은 지속적인 교육을 받을 수 있도록 몇 가지 자료를
제공합니다.

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

지속적인 교육 자료

Oracle Database 11g 작업에 대한 추가 정보는 다음 자료에서 찾아 볼 수 있습니다.

- Oracle University 웹 사이트
- Oracle University Knowledge Center
- Oracle Technology Network:
 - Oracle by Example
 - Oracle Magazine
 - Oracle Database 제품 페이지
- 기술 지원: My Oracle Support
- Oracle Database 제품 페이지

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle University



오라클에서 오라클을 배우십시오! Oracle University보다 오라클 기술을 더 잘 배울 수 있는 곳은 없습니다.

- **전세계적인 교육 서비스**
- **수강생 만족도 100%**
- **고객의 요구에 가장 적합한 학습 형식**
 - 정규 교육
 - 라이브 웹 클래스
 - 자가 학습 CD-ROM
- **자격증**

Oracle University

<http://www.oracle.com/education>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle University

Oracle University는 전세계에 교육 센터를 둔 세계에서 가장 큰 법인 교육 기관입니다. 목표는 수강생 만족도 100%입니다.

오라클 자격증은 IT 전문가 및 고용주가 작업자의 역량을 판단하는 데 중요한 기준으로 고려될 수 있는 업계에 널리 알려진 확실한 인증 자격입니다. 예를 들어, DBA는 다음과 같은 여러 가지 자격증 획득 방법이 있습니다.

- Oracle Certified Associate(OCA)
- Oracle Certified Professional(OCP)
- Oracle Certified Master(OCM) 및
- Oracle 10g: Managing Oracle on Linux Certified Expert와 같은 전문 자격증

지속적인 교육

- 데이터베이스 전문 과정:
 - *Oracle Database 10g: Administration Workshop II*
 - *Oracle Database 11g: Performance Tuning*
 - *Oracle Database 11g: Security*
- 기타 전문 과정:
 - *Oracle 11g: RAC and Grid Foundation Administration*
 - *Oracle Database 11g: RAC Administration*

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

지속적인 교육

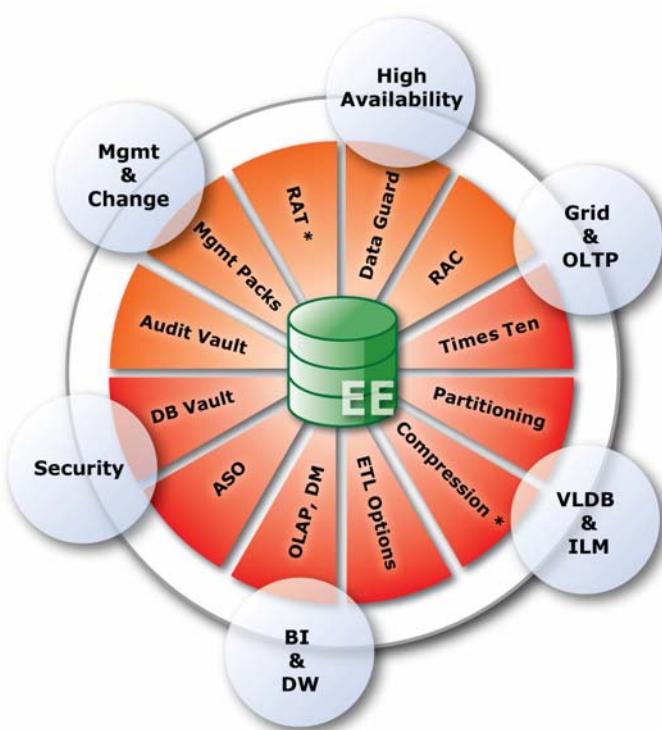
Oracle Database 10g: Administration Workshop II 과정 역시 데이터베이스 관리자가 되기 위한 교육 과정으로서 고급 데이터베이스 Recovery 방법, 성능 모니터 및 튜닝과 분산 데이터의 개념을 배우게 됩니다. 전문 과정을 시작하기 전에 이 과정을 이수하는 것이 좋습니다. 다음과 같은 추가 과정은 오라클 데이터베이스 관리자로서의 능력을 향상시킬 수 있는 여러 주제들로 제공합니다.

- *Oracle Database 11g: Performance Tuning*
- *Oracle Database 11g: Security*

기타 전문 과정으로는 *Oracle 11g: RAC and Grid Foundation Administration* 및 *Oracle Database 11g: RAC Administration* 있습니다.

모든 과정에 대한 최신 리스트를 확인하려면 Oracle University 웹 사이트를 참조하십시오. 귀하의 직무 내용에 해당하는 추천 교육 과정을 자세히 보려면 Oracle University Learning Paths를 참조하십시오.

데이터베이스 전문 영역



현대 엔터프라이즈 그리드

- Real Application Cluster
- 관리 팩
- TimesTen In-Memory 데이터베이스

Information Lifecycle Management

- Partitioning
- Advanced Compression

데이터 웨어하우징

- Oracle Information Appliances
- OLAP, Mining, Warehouse Builder

제어, 위험 및 규정 준수

- 보안 옵션
- Total Recall

변경 관리

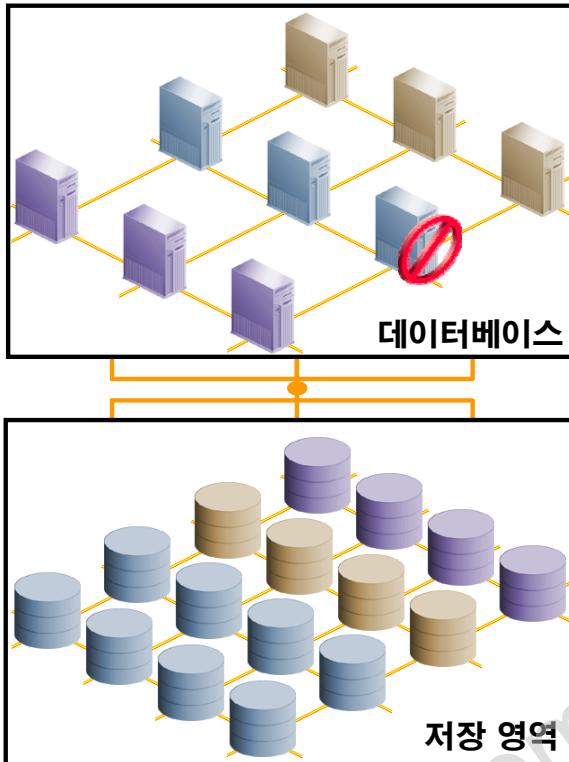
- Real Application Testing

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Real Application Clusters

- 여러 작업 로드를 단일 그리드로 통합
- 정보 플랫폼 가상화
- 유연한 물리적 Infrastructure(Dedicated Server 포함)



ORACLE

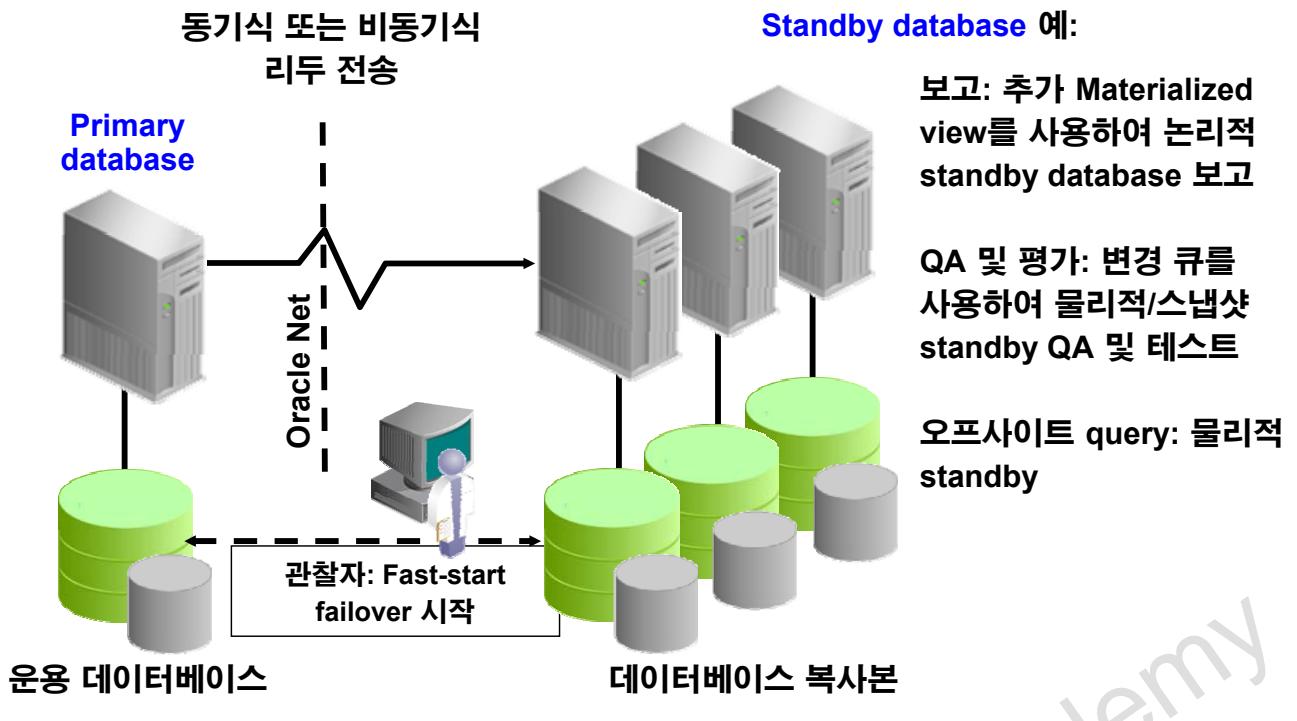
Copyright © 2010, Oracle. All rights reserved.

RAC 사용 이유

Oracle RAC(Real Application Clusters)를 사용하면 블레이드 등의 가격이 저렴한 표준 모듈식 서버 클러스터를 효율적으로 활용할 수 있습니다. RAC는 서비스에 대해 자동 작업 로드 관리 기능을 제공합니다. 서비스는 응용 프로그램 작업 로드에 해당하는 비즈니스 구성 요소로 구성된 응용 프로그램 그룹 또는 분류입니다. RAC의 서비스를 사용하면 데이터베이스 작업을 중단 없이 지속적으로 수행할 수 있으며, 여러 Instance에서 여러 서비스를 지원할 수 있습니다. 하나 이상의 Instance에서 실행되도록 서비스를 할당하고 대체 Instance가 백업 Instance 역할을 하도록 지정할 수 있습니다. Primary Instance가 실패하면 Oracle은 서비스를 실패한 Instance에서 정상적인 대체 Instance로 이동합니다. 또한 Oracle은 서비스를 호스팅하는 Instance 간의 연결에 대한 로드 밸런싱을 자동으로 수행합니다.

RAC는 여러 대의 저렴한 컴퓨터에서 제공되는 기능을 활용하여 데이터베이스 처리를 위한 대규모 단일 컴퓨터 역할을 하며, 모든 유형의 응용 프로그램에 대해 대규모 SMP 제품에 대한 대안으로 사용할 수 있는 유일한 대체 수단입니다. 표준 디스크 구조를 기반으로 하는 RAC는 클러스터의 서버 간에 인위적으로 데이터를 Partition하지 않고도 요청 시에 확장 및 축소(shrink) 할 수 있습니다. 또한 RAC에서는 단일 버튼을 통해 클러스터에 서버를 추가 및 제거할 수 있습니다. 그러므로 간편하게 데이터베이스에 서버를 제공하거나 데이터베이스에서 서버를 제거할 수 있습니다.

Oracle Data Guard



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Data Guard

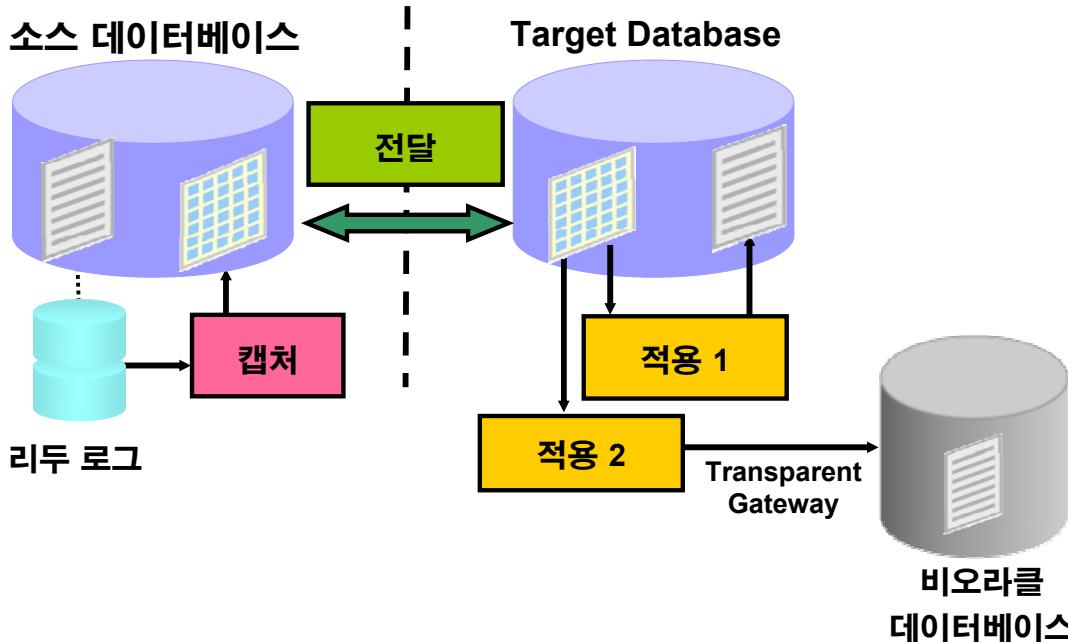
Oracle Data Guard는 관리, 모니터 및 자동화 소프트웨어 Infrastructure로, 운용 중인 데이터베이스 및 하나 이상의 standby database와 연동하여 데이터베이스 손상을 초래할 수 있는 failure, 오류 및 훼손으로부터 데이터를 보호해 줍니다. 이 도구는 Data Guard 구성에서 데이터베이스와 다른 구성 요소의 생성, 관리 및 모니터를 자동화하는 기능을 제공함으로써 중요한 데이터를 보호합니다. 또한 운용 중인 오라클 데이터베이스가 일상적인 유지 관리를 위해 오프라인으로 설정되거나 손상된 경우에 사용할 수 있도록, 운용 중인 데이터베이스의 복사본(standby database)을 유지 관리하는 프로세스를 자동화합니다.

Data Guard 구성에서는 운용 중인 데이터베이스를 *primary database*라고 합니다. *standby database*는 *primary database*의 동기화된 복사본입니다. *Primary database*의 백업 복사본을 사용하여 한 개에서 아홉 개의 *standby database*를 생성할 수 있습니다. Data Guard 구성은 *standby database*와 *primary database*로 이루어집니다. 각 *standby database*는 하나의 *primary database*와만 연관됩니다.

주: Cascaded Redo Log Destinations 기능을 사용하여 구성에 10개 이상의 *standby database*를 지정할 수 있습니다.

Data Guard 구성에서는 룰 반전을 지원하기 위한 *primary database*를 비롯하여 모든 *standby database*에서 *standby* 리두 로그 파일을 구성하는 것이 좋습니다.

Streams 개요



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Streams 개요

스트림은 데이터베이스 내에서 또는 데이터베이스 간에 발생하는 정보의 흐름입니다. Oracle Streams는 데이터 스트림을 통해 데이터와 메시지를 공유하는 데 사용할 수 있는 일련의 프로세스 및 데이터베이스 구조입니다. stream에 배치되는 정보의 단위를 이벤트라고 합니다.

- LCR로 형식이 지정된 DDL 또는 DML 변경 사항
- 유저 생성 이벤트

이벤트는 큐 간에 스테이지되고 전달됩니다.

Streams는 대부분의 경우 플랫폼 또는 릴리스에 관계없이 모든 데이터베이스가 갱신 가능한 복제본(replication)으로 간주됩니다. 특성은 다음과 같습니다.

- 모든 사이트: 활성 및 갱신 가능
- 자동 충돌 감지 및 선택적 해결
- 데이터 변형 지원
- 유연한 구성: n-way, 허브 앤드 스포크(Hub-and-Spoke) 등
- 다른 데이터베이스 플랫폼, 릴리스 및 스키마
- 응용 프로그램에 대한 높은 가용성 제공(갱신 충돌 방지 또는 관리 가능)

Oracle Streams: 기본 요소

Oracle Streams를 사용하면 데이터베이스 내에서나 데이터베이스 간에 데이터 스트림의 데이터 및 이벤트를 공유할 수 있습니다.

Oracle Streams에서는 큐를 사용하여 전달 또는 소비를 위한 이벤트를 스테이지합니다. Oracle Streams을 사용하여 큐 간에 이벤트를 전달하고, 이 큐를 동일 데이터베이스 또는 서로 다른 데이터베이스에 배치할 수 있습니다. Streams에서 사용된 큐에서 두 가지 이벤트 유형인 캡처된 이벤트(논리적 변경 레코드 또는 LCR) 및 유저 enqueue 이벤트(메시지 또는 LCR일 수 있음)를 스테이지할 수 있습니다.

- 데이터베이스에 대한 변경 사항은 리두 로그에서 캡처될 수 있습니다. 그런 다음 이러한 변경 사항의 형식을 LCR로 지정할 수 있습니다. LCR은 DML(데이터 조작어) 또는 DDL(데이터 정의어) 변경 사항을 나타낼 수 있습니다. 리두 로그에서 변경 사항이 생성된 데이터베이스는 소스 데이터베이스라고 합니다.
- 또한 유저 이벤트를 유저 응용 프로그램에서 명시적으로 enqueue할 수도 있습니다. 이렇게 명시적으로 enqueue된 이벤트는 LCR 또는 유저 생성 메시지일 수 있습니다. 메시지는 큐에 삽입되거나 큐에서 검색되는 정보의 최소 단위입니다. 메시지는 데이터는 물론 메시지 데이터의 해석 및 사용을 제어하는 정보로 구성되어 있습니다.

Oracle Streams를 일련의 작은 작업으로 구분할 수 있습니다. 이러한 작업을 구성하면 스트림에 유입되는 정보, 스트림이 노드 간에 유입되는 방법, 메시지가 각 노드로 유입될 때 해당 이벤트에 발생하는 상황, 스트림이 종료되는 방법을 제어할 수 있습니다.

또한 각 작업을 커스터마이즈하여 특정 요구 사항 및 업무 요구를 해결할 수 있습니다. 그 결과 이전 솔루션보다 뛰어난 기능 및 유연성을 제공하는 새 기능을 활용하여 이벤트를 캡처 및 관리하고 다른 데이터베이스 및 응용 프로그램과 이벤트를 공유할 수 있습니다. Oracle Streams는 분산된 기업 및 응용 프로그램, 데이터 웨어하우스 및 고가용성 솔루션을 구축 및 운용하는데 필요한 기능을 제공합니다.

Oracle Streams의 세 가지 기본 작업은 다음과 같습니다.

- 캡처:** 리두 로그에서 DML 또는 DDL 이벤트를 자동으로 캡처합니다. 유저 생성 이벤트는 자동으로 캡처되지 않지만 명시적인 enqueue 작업을 통해 큐로 배치됩니다.
- 스테이지:** 데이터베이스 간에 이벤트를 저장 및 전달합니다. 전달은 필요한 경우 명시적으로 수행할 수 있습니다.
- 적용:** DML 또는 DDL 이벤트를 대상 데이터베이스에 적용하거나 이벤트를 응용 프로그램에 전달합니다.

이러한 작업을 단일 데이터베이스에서 수행하거나 다른 데이터베이스의 작업들과 조합하여 분산 환경을 구성할 수 있습니다.

다중 데이터베이스 Streams

이벤트는 각 데이터베이스의 스테이지 영역 간에 전달됩니다. 캡처 및 소비 요소는 모든 데이터베이스에서 활성화할 수 있습니다. 예를 들어, 캡처 프로세스, 전달 작업에서 양방향 데이터 복제(replication)를 구성하고 각 사이트에 프로세스를 적용할 수 있습니다. 또는 한 사이트에서 캡처 및 전달을 담당하는 단일 소스 시스템을 구성하고 이를 다른 여러 데이터베이스에 적용할 수 있습니다. 또한 데이터베이스 수도 임의로 지정할 수 있습니다. 좀 더 복잡한 환경에서는 수백 개의 데이터베이스가 Oracle Streams와 정보를 공유해야 할 수 있습니다.

Oracle Technology Network

Oracle Technology Network는 데이터베이스, Application Server, Collaboration Suite 및 개발 도구를 포함한 핵심 Oracle 소프트웨어 제품에 대한 정보를 보유하는 무료 자료입니다. 다음 사항에 액세스할 수 있습니다.

- 기술 센터
- 토론 포럼
- 소프트웨어 다운로드
- 온라인 설명서
- Oracle by Example
- 코드 예제

... 이외에도 많은 자료와 정보가 제공됩니다.



<http://www.oracle.com/technology>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Technology Network

OTN(Oracle Technology Network)은 오라클 기술 및 제품에 대한 최신 소식을 제공합니다. 또한 OTN은 피어 투 피어 포럼, 백서, 보안 게시판 및 오라클 전문가에게 필요한 기타 중요한 정보를 제공합니다.

Oracle 소프트웨어를 잘 활용하기 위한 참고 사항과 기술 외에 OTN에서는 Oracle 소프트웨어도 다운로드할 수 있습니다. 모든 소프트웨어 다운로드는 무료이며 해당 응용 프로그램을 개발하고 프로토 타입을 만드는 동안에만 제품을 사용할 수 있는 개발 사용권을 각각의 유저에게 부여한다는 사실을 기억하십시오.

보안

The screenshot shows the Oracle Technology Network website's security section. At the top, there's a navigation bar with links for 'shortcuts', 'GETTING STARTED', 'DOWNLOADS', 'DOCUMENTATION', 'FORUMS', 'ARTICLES', 'SAMPLE CODE', and 'TUTORIALS'. A search bar and a 'Technology Network' dropdown are also present. On the left, a sidebar lists categories like 'PRODUCTS', 'TECHNOLOGIES', 'ARCHITECTURE', and 'COMMUNITY'. The main content area features a 'Security Technology Center' section with a lock icon, followed by a 'What's New' section containing news items about Oracle's security innovations and standards.

<http://www.oracle.com/technology/deploy/security/index.html>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

보안

데이터베이스 보안 관련 측면에 대한 자세한 정보를 보려면 정기적으로 개선되는 "Security Technology Center"를 방문하십시오.

Oracle by Example

- **OBE란?**
 - 실전 중심의 단계별 교육
- **참조 웹 사이트**
 - <http://www.oracle.com/technology/obe>
- **제공되는 리소스**
 - 여러 오라클 제품 영역에서 수백 개의 OBE 자습서 제공

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

Oracle by Example

OBE(Oracle by Example) 시리즈는 Oracle 제품의 여러 새로운 기능을 사용하는 방법에 대한 실전 중심의 단계별 교육을 제공하므로, 신제품의 기능을 학습하는 데 걸리는 시간을 줄이면서 유저 환경에서 이 기능을 구현하는 방법에 대한 이해를 높일 수 있습니다. 현재 OBE는 Oracle database, Fusion Middleware, Oracle Application Server, Oracle Enterprise Manager Grid Control, Oracle Collaboration Suite, JDeveloper 및 Business Intelligence에서 사용할 수 있습니다. OBE는 <http://www.oracle.com/technology/obe>에서 액세스할 수 있습니다.

Oracle Magazine

- 무료 구독

- Oracle Magazine Archives

<http://www.oracle.com/technology/oramag/index.html>



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Magazine

Oracle Magazine은 OTN을 통해 액세스할 수 있는 여러 유형의 리소스 중 하나입니다.
무료 구독권을 메일로도 받을 수 있습니다.

Oracle Applications 커뮤니티

Oracle Technology Network는 Oracle Applications 유저 및 구현자를 위한 자료입니다. 다음 사항에 액세스할 수 있습니다.

- 토론 포럼
- 유저 그룹
- 온라인 채팅
- 설명서
- 교육
- 업그레이드 정보
 - 이외에도 많은 자료와 정보가 제공됩니다.



<http://www.oracle.com/technology/community/apps/index.html>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy
Use Only

기술 지원: My Oracle Support

My Oracle Support 사용권은 연간 지원 유지 관리비에 포함됩니다.
My Oracle Support는 최신 기술 정보뿐만 아니라 다음에 대한
액세스를 제공합니다.

- 서비스 요청(SR)
- 자격증 정보
- 오라클 전문가와 함께 하는 기술 포럼
- 소프트웨어 패치
- 버그 보고



<http://metalink.oracle.com>



Copyright © 2010, Oracle. All rights reserved.

Oracle MetaLink

My Oracle Support는 오라클의 기술 지원이 제공되는 통로입니다. My Oracle Support는 많은 문제를 해결할 수 있는 자료뿐만 아니라 오라클 관리자와 개발자가 겪는 공통적인 문제에 대한 해답도 찾아볼 수 있습니다.

Oracle Technology Network와 마찬가지로 My Oracle Support는 오라클 전문가에게 영향을 미칠 수 있는 주요 문제들에 대한 최신 정보를 제공합니다.

Oracle Database 제품 페이지

OTN 상의 Oracle Database 제품 페이지 링크

- 데이터베이스 핵심 영역
- 백서
- 데이터 시트 옵션
- 관련 기술
- 토론
- 기타 유용한 리소스

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Database 제품 페이지

OTN 상의 Oracle Database 제품 페이지는 다음과 같은 유용한 정보와 링크되어 있습니다.

- **데이터베이스 핵심 영역:** 그리드/클러스터링, 고가용성, 성능 및 관리 효율성과 같은 다른 데이터베이스와 관련된 링크를 제공합니다.
- **백서:** Oracle Database 11g 제품군에 대한 추가 내용을 학습하는데 도움이 되는 자세한 정보를 제공합니다.
- **데이터 시트 옵션:** 다양한 Oracle Database 11g 옵션에 대한 자세한 정보를 제공합니다.
- **관련 기술:** Oracle Application Express 및 Oracle SQL Developer와 같은 관련 기술에 대한 링크를 제공합니다.
- **토론:** Oracle Database 11g에 관련된 지속적인 토론 내용들에 대한 링크를 제공합니다.
- **기타 리소스:** 설명서, 포럼, 추가 학습에 대한 링크 및 기타 유용한 리소스에 대한 링크를 제공합니다.

여기에서 제공되는 문서는 필요에 따라 언제든지 갱신되는 라이브 문서이며, 언제든 새로운 문서가 추가될 수 있습니다. 사용 가능한 최신 정보를 찾을 수 있도록 정기적으로 확인하시기 바랍니다.

감사의 말

Oracle University가 귀하에게 도움이 되었기를 바랍니다.
고객의 요구를 만족시킬 수 있는 더 좋은 서비스를 위해
귀하의 의견을 기다립니다.

- 종강 평가
- Oracle University CS(고객 만족도) 부서
- 오라클 교육 서비스

다시 뵙기를 바랍니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

감사의 말

Oracle University의 목표는 오라클 기술 사용 능력을 향상시키는 것이며 고객의 요구 사항에 시기 적절하게 부합하는 정보를 제공하며 함께 협력하는 것입니다.

더 좋은 서비스를 위해 잠시만 시간을 내어 종강 평가를 작성해 주십시오. 미국의 경우 아래 CS 부서로 전자 메일을 보내 주십시오.

customersat_us@oracle.com

오라클 교육을 계속하는 데 있어 질문이 있거나, 적절한 과정을 찾는 데 도움이 필요하거나, 귀사에서 직접 교육을 받고 싶은 경우에는 오라클 교육 서비스 담당자에게 문의하십시오.

미국 내 전화 번호는 800.529.0165입니다. 미국 이외의 국가에서는 다음 웹 사이트를 방문하십시오.

<http://www.oracle.com/education/index.html?contact.html>

감사합니다. 다시 뵙기를 바랍니다.