

Oracle Database 11g: Administration Workshop I(한글판)

블룸 I • 학생용

D50102KR20

Edition 2.0

2010년 5월

D67248

ORACLE®

1

오라클 데이터베이스 구조 탐색

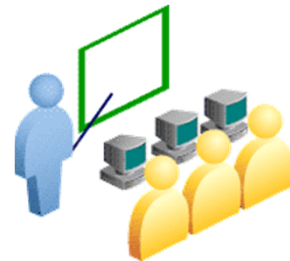
ORACLE®

Copyright © 2010, Oracle. All rights reserved.

목표

이 단원을 마치면 다음을 수행할 수 있습니다.

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

목표

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

오라클 데이터베이스

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



ORACLE

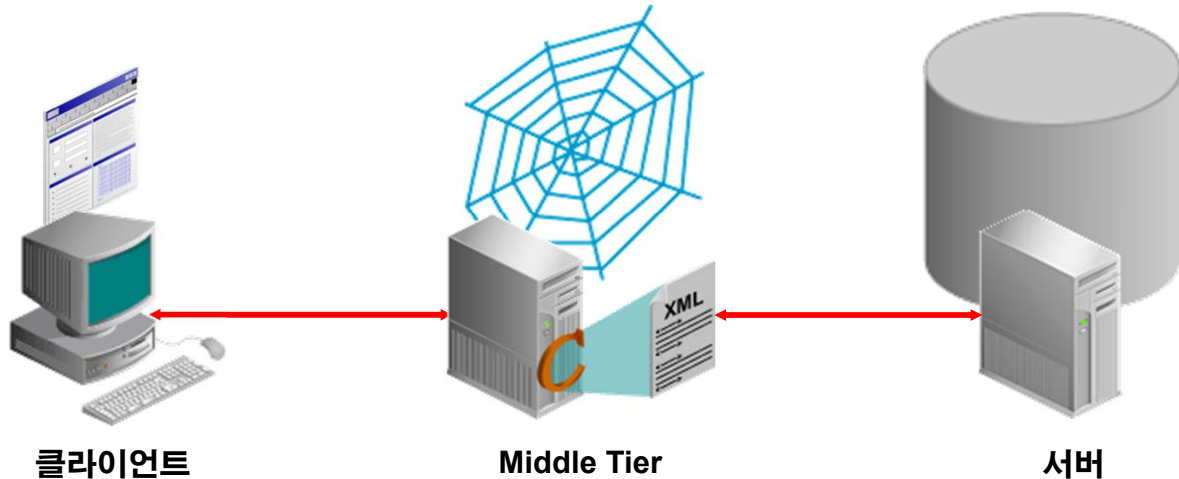
Copyright © 2010, Oracle. All rights reserved.

오라클 데이터베이스

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

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

서버에 연결



여러 계층 구조를 보여주는 그림

ORACLE

Copyright © 2010, Oracle. All rights reserved.

서버에 연결

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

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

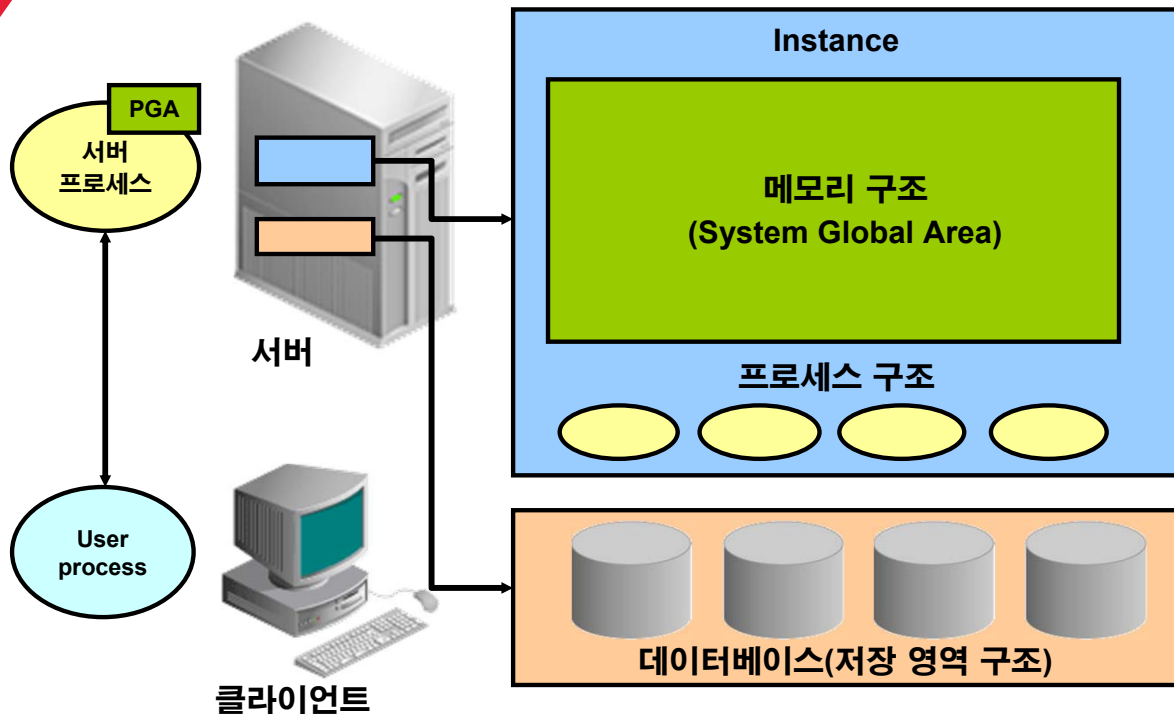
서버에 연결(계속)

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

일반적인 여러 계층 구조에는 다음과 같은 구성 요소가 포함됩니다.

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

오라클 데이터베이스 서버 구조: 개요



ORACLE

Copyright © 2010, Oracle. All rights reserved.

오라클 데이터베이스 서버 구조

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

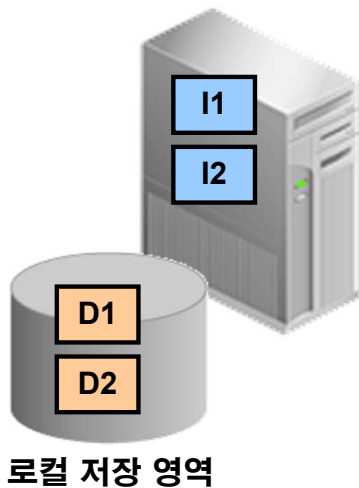
데이터베이스에는 물리적 구조와 논리적 구조로 구성됩니다. 물리적 구조와 논리적 구조는 서로 별개의 구조이기 때문에 논리적 저장 영역 구조에 대한 액세스에 영향을 주지 않고 데이터의 물리적 저장 영역을 관리할 수 있습니다.

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

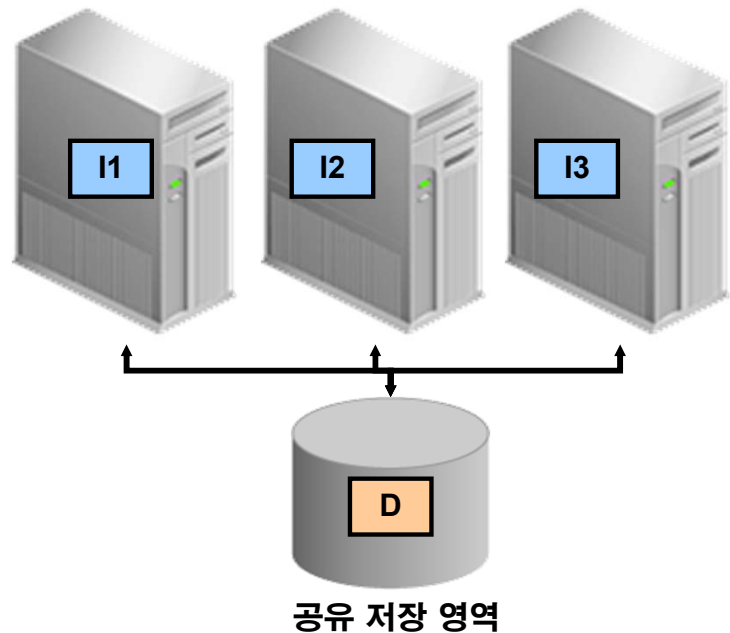
주: Oracle 자동 저장 영역 관리(ASM)는 메모리 및 프로세스 구성 요소에 대해 Instance의 개념을 사용하지만 특정 데이터베이스와 연관되지는 않습니다.

Instance: 데이터베이스 구성

클러스터화되지 않은 시스템



클러스터화된 시스템



ORACLE

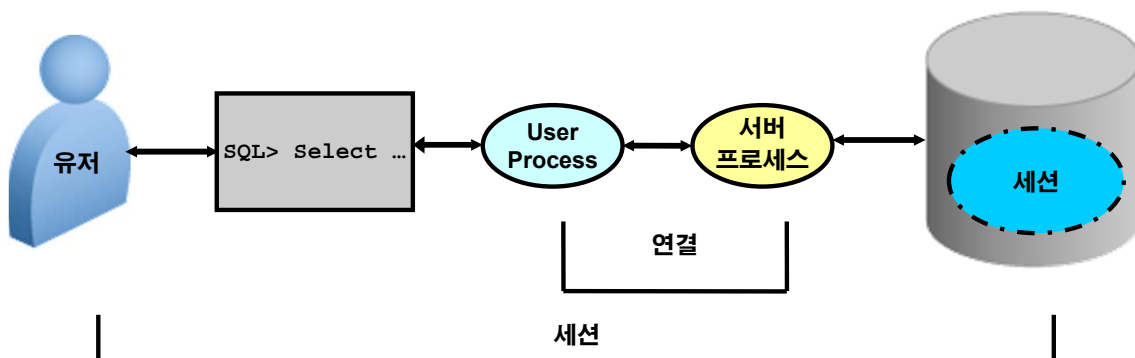
Copyright © 2010, Oracle. All rights reserved.

Instance: 데이터베이스 구성

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

데이터베이스 Instance에 연결

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 Instance에 연결

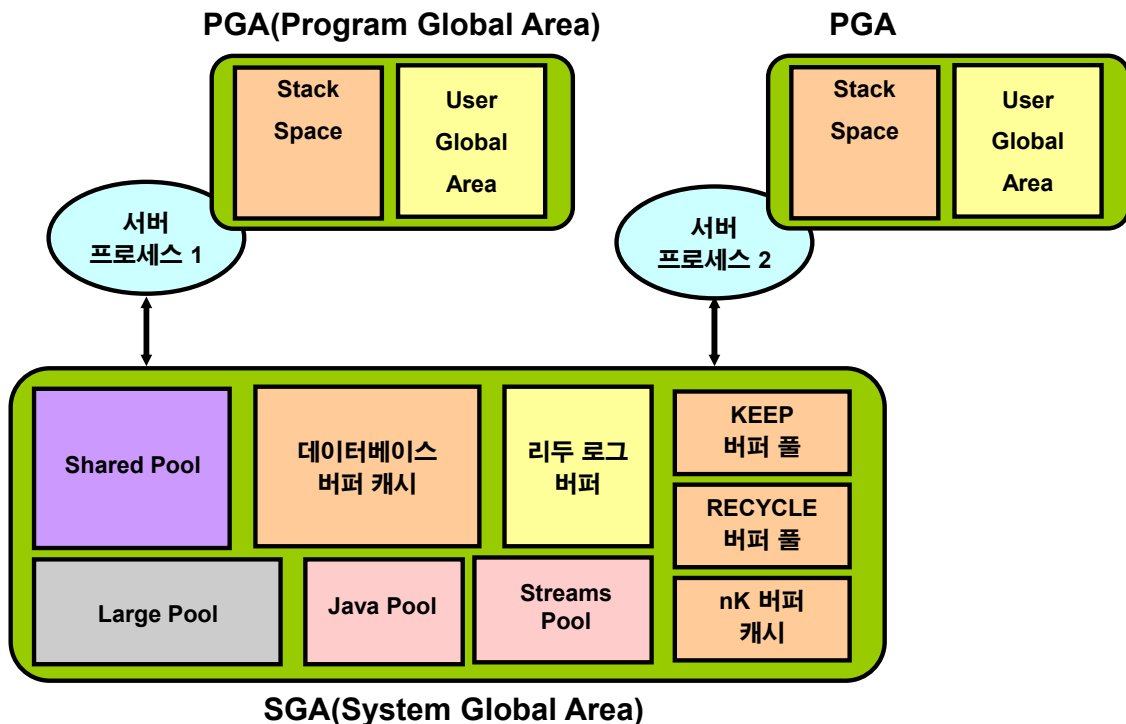
연결 및 세션은 User Process와 밀접하게 연관되어 있지만 그 의미는 전혀 다릅니다.

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

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

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

오라클 데이터베이스 메모리 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

오라클 데이터베이스 메모리 구조

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

Instance에는 다음과 같은 두 가지 기본 메모리 구조가 연관되어 있습니다.

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

오라클 데이터베이스 메모리 구조(계속)

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

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

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

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

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

오라클 데이터베이스는 초기화 파라미터를 사용하여 메모리 구조를 생성 및 관리합니다.

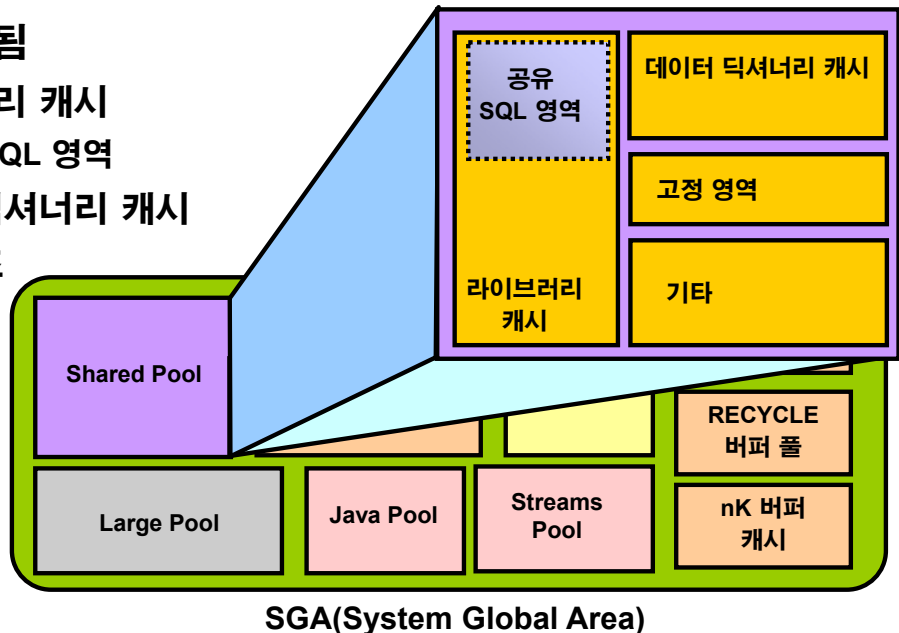
메모리를 관리하는 가장 간단한 방법은 데이터베이스가 메모리를 자동으로 관리

및 튜닝하도록 하는 것입니다. 그러려면 대부분의 플랫폼에서 목표 메모리 크기 초기화

파라미터(MEMORY_TARGET)와 최대 메모리 크기 초기화 파라미터(MEMORY_MAX_TARGET)만 설정하면 됩니다.

Shared Pool

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Shared Pool

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

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

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

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

Shared Pool(계속)

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

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

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

PL/SQL 프로그램 단위에 포함된 개별 SQL 문은 다른 SQL 문과 같은 방식으로 처리됩니다.

PL/SQL 프로그램 단위의 원래 값과는 달리 이러한 SQL 문은 공유 영역을 사용하여 구문 분석된 표현을 보관하고 명령문을 실행하는 각 세션에 대해 전용(Private) 영역을 사용합니다.

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

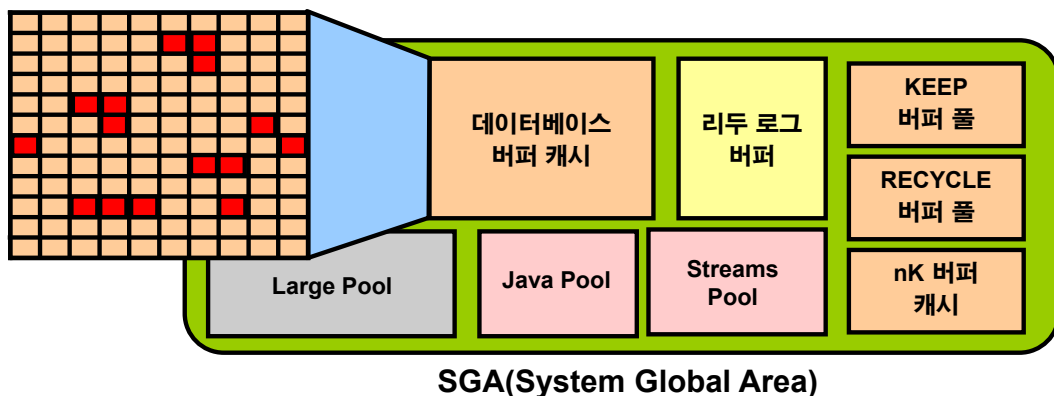
Query 결과 및 Query 부분은 SQL Query 결과 캐시의 메모리에서 캐시에 저장될 수 있습니다. 그러면 데이터베이스가 캐시된 결과를 사용하여 이러한 Query 및 Query 부분의 향후 실행을 수행할 수 있습니다. SQL query 결과 캐시에서 결과를 검색하는 것이 Query를 재실행하는 것보다 빠르기 때문에, 자주 실행하는 Query의 경우 해당 결과가 캐시에 저장되면 성능이 크게 향상됩니다.

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

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

데이터베이스 버퍼 캐시

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 버퍼 캐시

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

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

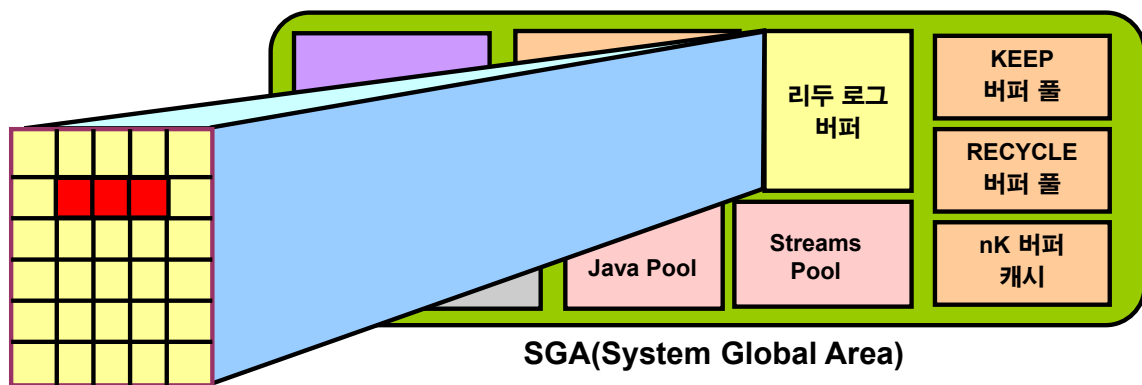
캐시의 버퍼는 LRU(Least Recently Used) List와 접근 횟수의 조합을 사용하는 복합 알고리즘에 의해 관리됩니다. LRU는 디스크 액세스를 최소화하기 위해 최근에 사용된 블록이 메모리에 유지되도록 보장합니다.

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

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

리두 로그 버퍼

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

리두 로그 버퍼

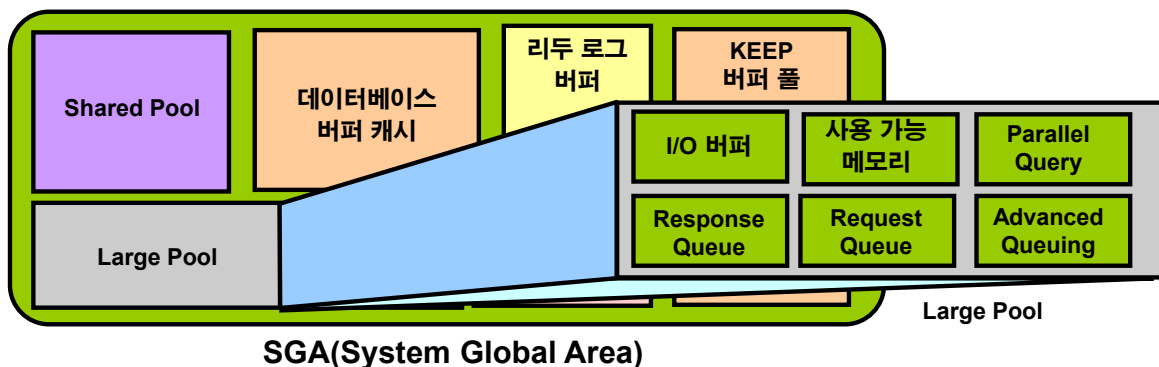
리두 로그 버퍼는 데이터베이스에 대한 변경 사항 관련 정보가 포함된 SGA의 순환 버퍼입니다. 이 정보는 리두 항목에 저장됩니다. 리두 항목은 DML, DDL 또는 내부 작업에 의해 데이터베이스에 수행된 변경 사항을 재생성(또는 리두)하는 데 필요한 정보를 포함합니다. 필요한 경우 리두 항목은 데이터베이스 Recovery에 사용됩니다.

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

Large Pool

다음을 위한 대규모 메모리 할당을 제공합니다.

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Large Pool

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

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

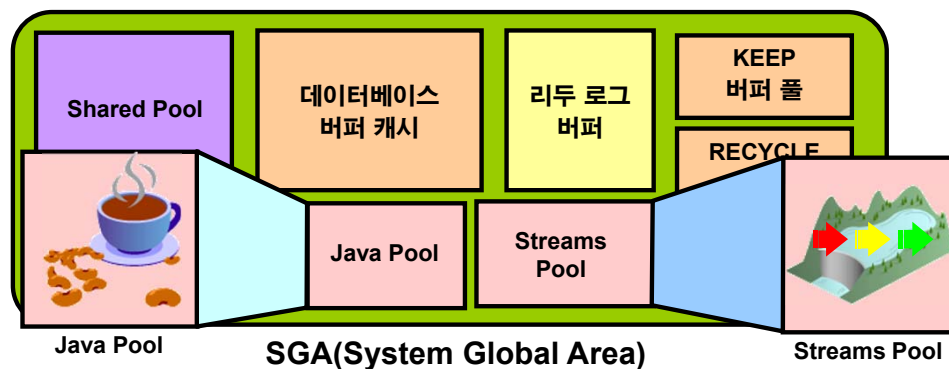
Shared Server, Oracle XA 또는 Parallel Query 버퍼에 대해 Large Pool에서 세션 메모리를 할당함으로써 오라클 데이터베이스는 기본적으로 공유 SQL을 캐시에 저장하는 데 Shared Pool을 사용할 수 있으며, 공유 SQL 캐시를 축소(Shrink)하면 발생하는 성능 오버헤드를 방지할 수 있습니다.

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

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

Java Pool 및 Streams Pool

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Java Pool 및 Streams Pool

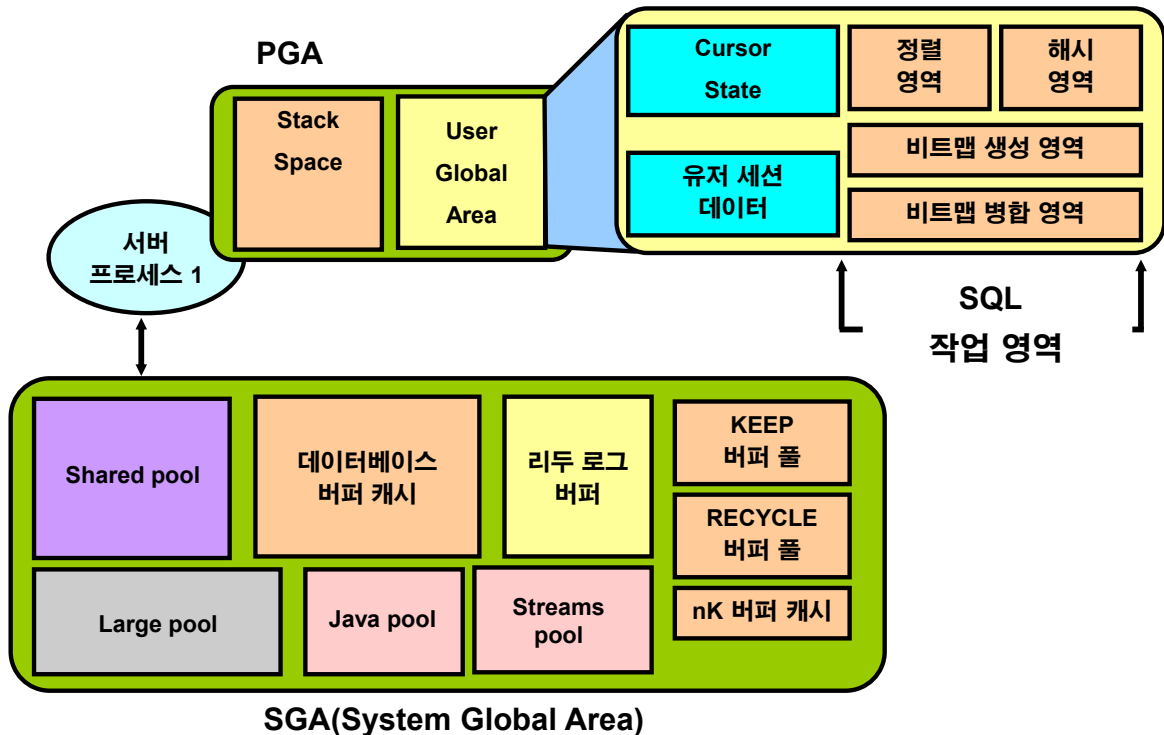
Java Pool 메모리는 JVM의 모든 세션별 Java 코드 및 데이터를 저장하는 데 사용되며, 오라클 데이터베이스를 실행 중인 모드에 따라 다른 방식으로 사용됩니다.

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

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

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

PGA(Program Global Area)



ORACLE

Copyright © 2010, Oracle. All rights reserved.

PGA(Program Global Area)

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

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

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

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

퀴즈

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

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

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 2

퀴즈

데이터 파일에서 데이터베이스 버퍼 캐시로 읽혀지는 것은 무엇입니까?

1. 행
2. 변경 사항
3. 블록
4. SQL

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 3

프로세스 구조

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

ORACLE

Copyright © 2010, Oracle. All rights reserved.

프로세스 구조

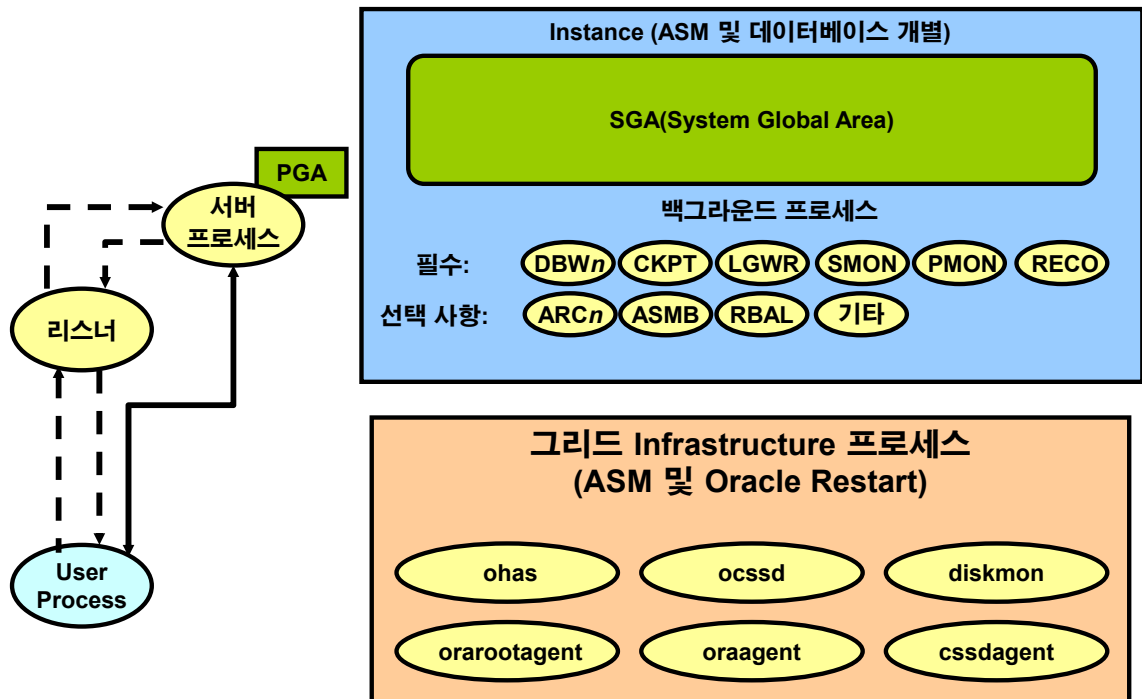
오라클 데이터베이스 시스템의 프로세스는 세 가지 주요 그룹으로 나눌 수 있습니다.

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

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

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

프로세스 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

프로세스 구조

서버 프로세스

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

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

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

백그라운드 프로세스

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

프로세스 구조(계속)

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

- DBWn(데이터베이스 기록자 프로세스)
- LGWR(로그 기록자 프로세스)
- CKPT(체크포인트 프로세스)
- SMON(시스템 모니터 프로세스)
- PMON(프로세스 모니터 프로세스)
- RECO(복구자 프로세스)
- CJQ0(조정자 작업 큐)
- Jnnn(작업 슬레이브 프로세스)
- ARCn(아카이버 프로세스)
- QMNn(큐 모니터 프로세스)

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

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

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

Linux 및 UNIX 시스템의 Oracle 그리드 Infrastructure 프로세스에는 다음이 포함됩니다.

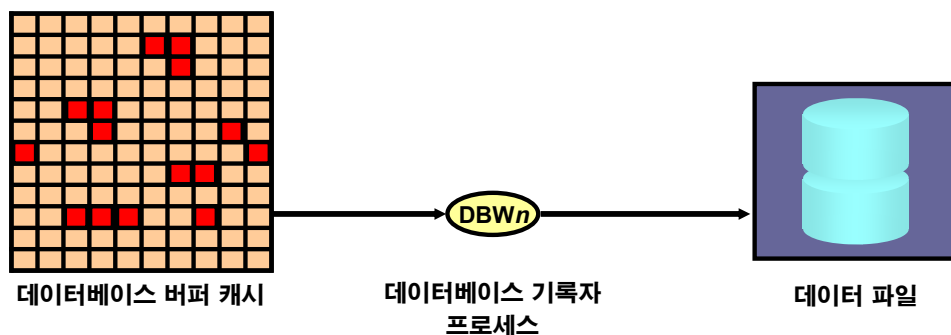
- ohasd: Oracle Clusterware 프로세스를 시작하는 Oracle High Availability Service Daemon
- ocssd: Cluster Synchronization Service Daemon
- diskmon: HP Oracle Exadata Storage Server에 대한 입력 및 출력 한도를 지정하는 Disk Monitor Daemon
- cssdagent: CSS Daemon인 ocssd를 시작 및 정지하고 상태 확인
- oraagent: 오라클 특정 요구 사항 및 복잡한 리소스를 지원하기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

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

DBWn(데이터베이스 기록자 프로세스)

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

- 다른 처리를 수행하는 동안 비동기적으로 기록
- 체크포인트 전진



ORACLE

Copyright © 2010, Oracle. All rights reserved.

DBWn(데이터베이스 기록자 프로세스)

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

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

DBWn(데이터베이스 기록자 프로세스)(계속)

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

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

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

DB_WRITER_PROCESSES 초기화 파라미터는 DBWn 프로세스 수를 지정합니다. 최대 DBWn 프로세스 수는 36개입니다. 시작 중에 사용자가 이 수를 지정하지 않으면 오라클 데이터베이스가 CPU 및 프로세스 그룹의 수에 따라 DB_WRITER_PROCESSES를 설정하는 방법을 결정합니다.

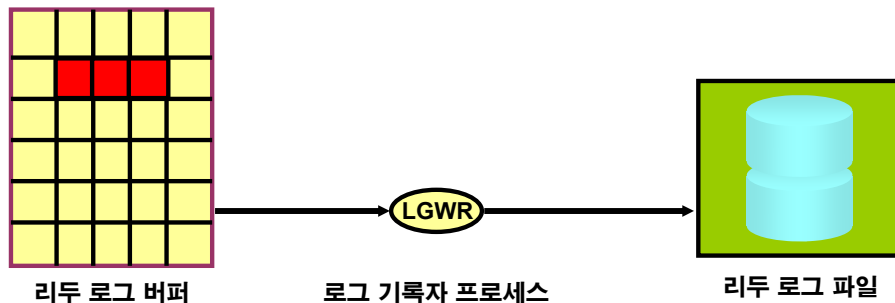
DBWn 프로세스는 다음과 같은 경우에 더티 버퍼를 디스크에 기록합니다.

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

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

LGWR(로그 기록자 프로세스)

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

LGWR(로그 기록자 프로세스)

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

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

LGWR는 다음 경우 기록합니다.

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

LGWR(LogWriter 프로세스) (계속)

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

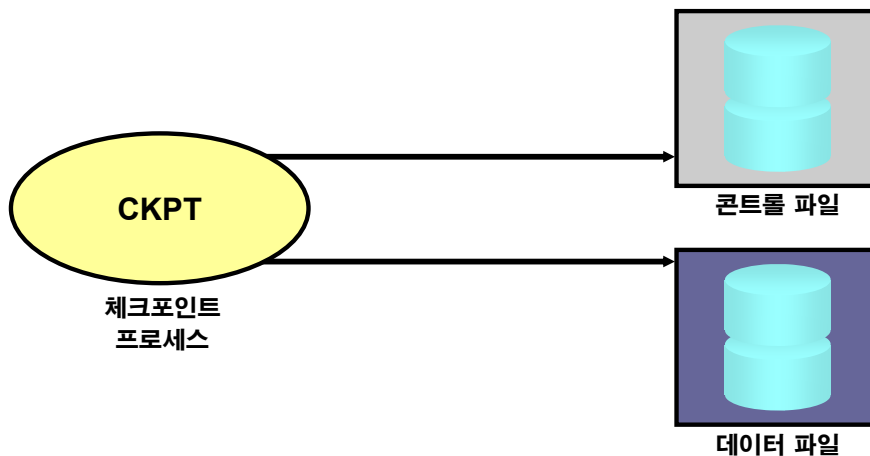
유저가 COMMIT 문을 발행하면 LGWR는 리두 로그 버퍼에 커밋 레코드를 배치하고 이를 트랜잭션의 리두 항목과 함께 디스크에 즉시 기록합니다. 데이터 블록에 대한 해당하는 변경 사항은 보다 효율적으로 기록될 수 있을 때까지 지연됩니다. 이를 빠른 커밋 메커니즘(*fast commit mechanism*)이라고 합니다. 트랜잭션의 커밋 레코드를 포함하는 리두 항목의 원자 쓰기는 트랜잭션의 커밋 여부를 확인하는 단일 이벤트입니다. 오라클 데이터베이스는 데이터 버퍼를 아직 디스크에 기록하지 않은 경우에도 커밋 트랜잭션에 대해 성공 코드를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

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

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

CKPT(체크포인트 프로세스)

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

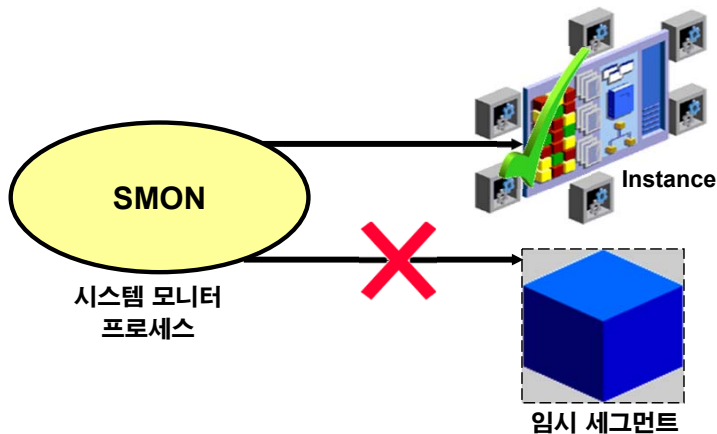
CKPT(체크포인트 프로세스)

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

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

SMON(시스템 모니터 프로세스)

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



ORACLE®

Copyright © 2010, Oracle. All rights reserved.

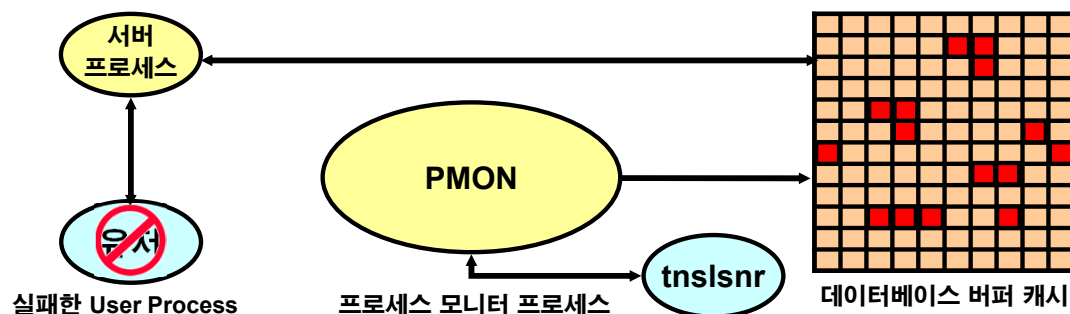
SMON(시스템 모니터 프로세스)

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

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

PMON(프로세스 모니터 프로세스)

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

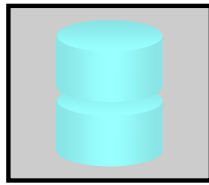
PMON(프로세스 모니터 프로세스)

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

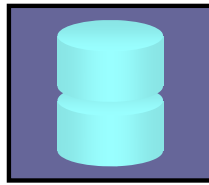
PMON은 정기적으로 디스패처와 서버 프로세스의 상태를 확인하여 실행이 정지된 경우(오라클 데이터베이스에서 의도적으로 종료한 경우 제외) 이를 재시작합니다. 또한 Instance 및 디스패처 프로세스에 대한 정보도 네트워크 리스너에 등록합니다.

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

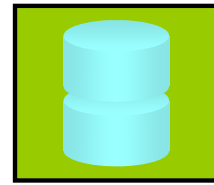
데이터베이스 저장 영역 구조



콘트롤 파일



데이터 파일



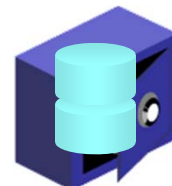
온라인 리두 로그 파일



파라미터 파일



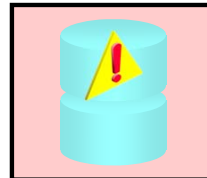
백업 파일



아카이브된 리두
로그 파일



Password File



Alert Log File 및 Trace File

ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 저장 영역 구조

오라클 데이터베이스는 다음과 같은 파일로 구성되어 있습니다.

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

다음은 성공적인 데이터베이스 실행에 필요한 추가 파일입니다.

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

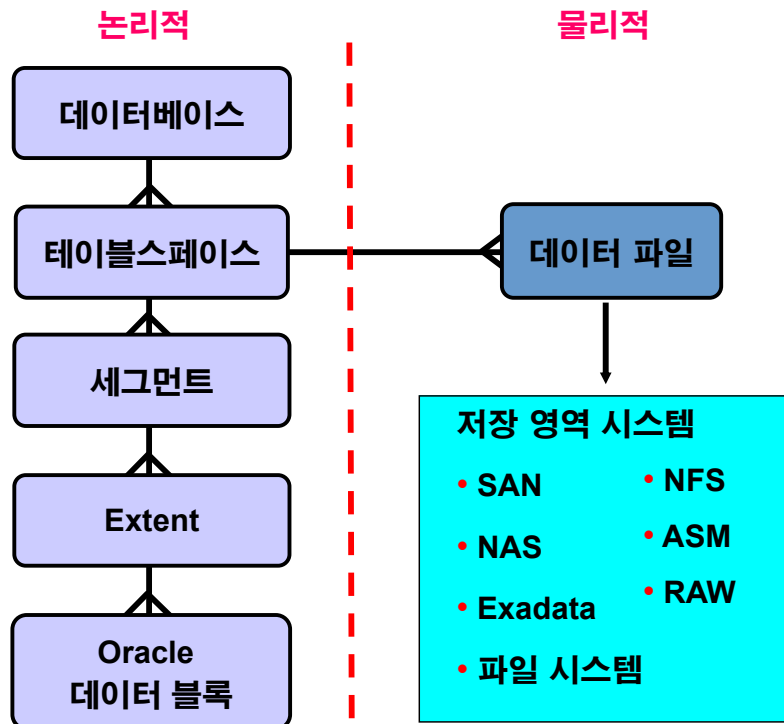
가능하게 합니다.

데이터베이스 저장 영역 구조(계속)

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

주: 파라미터 파일, Password File, Alert File 및 Trace File에 대해서는 다른 단원에서 설명합니다.

논리적 및 물리적 데이터베이스 구조



ORACLE

Copyright © 2010, Oracle. All rights reserved.

논리적 및 물리적 데이터베이스 구조

데이터베이스에는 논리적 구조와 물리적 구조가 있습니다.

데이터베이스, 테이블스페이스 및 데이터 파일

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

테이블스페이스

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

데이터 블록

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

논리적 및 물리적 데이터베이스 구조(계속)

Extent

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

세그먼트

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

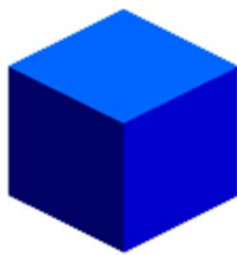
- **데이터 세그먼트:** 클러스터화되지 않은 각각의 비인덱스 구성(Non-index-organized) 테이블에는 데이터 세그먼트가 있습니다. 단, External Table, Global 임시 테이블(Temporary Table), Partition 테이블은 예외입니다. 이 경우 각 테이블에는 한 개 이상의 세그먼트가 있습니다. 테이블 데이터는 모두 해당 데이터 세그먼트의 Extent에 저장됩니다. Partition 테이블의 경우 각 Partition은 데이터 세그먼트를 가집니다. 각 클러스터는 데이터 세그먼트를 가집니다. 클러스터에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

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

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

세그먼트, Extent 및 블록

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



세그먼트



Extent



데이터 블록



디스크 블록
(파일 시스템 저장 영역)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

세그먼트, Extent 및 블록

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

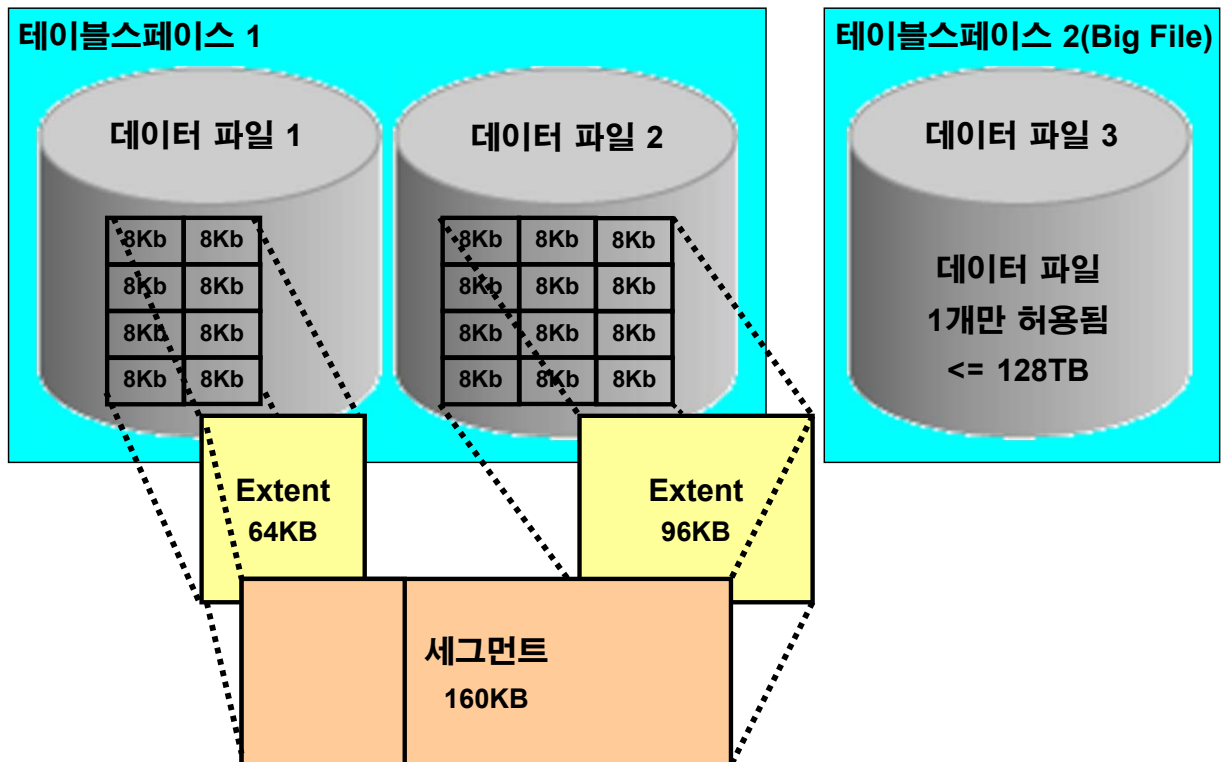
데이터베이스에서 OS(운영 체제)의 데이터 블록 집합을 요청하면 OS는 이를 저장 장치의 디스크 블록이나 실제 파일 시스템으로 매핑합니다. 따라서 데이터베이스에 있는 데이터의 물리적 주소를 알 필요가 없으며 데이터 파일을 여러 디스크에 스트라이핑 또는 Mirroring할 수도 있습니다.

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

읽기와 쓰기가 무작위로 발생하는 트랜잭션 응용 프로그램을 지원하는 데이터베이스인 경우에는 블록 크기를 작게 지정하는 것이 유용할 수도 있습니다. 최대 블록 크기는 OS에 따라 다릅니다. 최소 Oracle 블록 크기는 2KB인데 이렇게 작은 크기는 거의 사용되지 않습니다.

테이블스페이스의 블록 크기를 표준이 아닌 다른 값으로 설정할 수 있습니다. 자세한 사항은 *Oracle Database Administrator's Guide*를 참조하십시오.

테이블스페이스 및 데이터 파일



ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블스페이스 및 데이터 파일

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

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

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

SYSTEM 및 SYSAUX 테이블스페이스

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

ORACLE

Copyright © 2010, Oracle. All rights reserved.

SYSTEM 및 SYSAUX 테이블스페이스

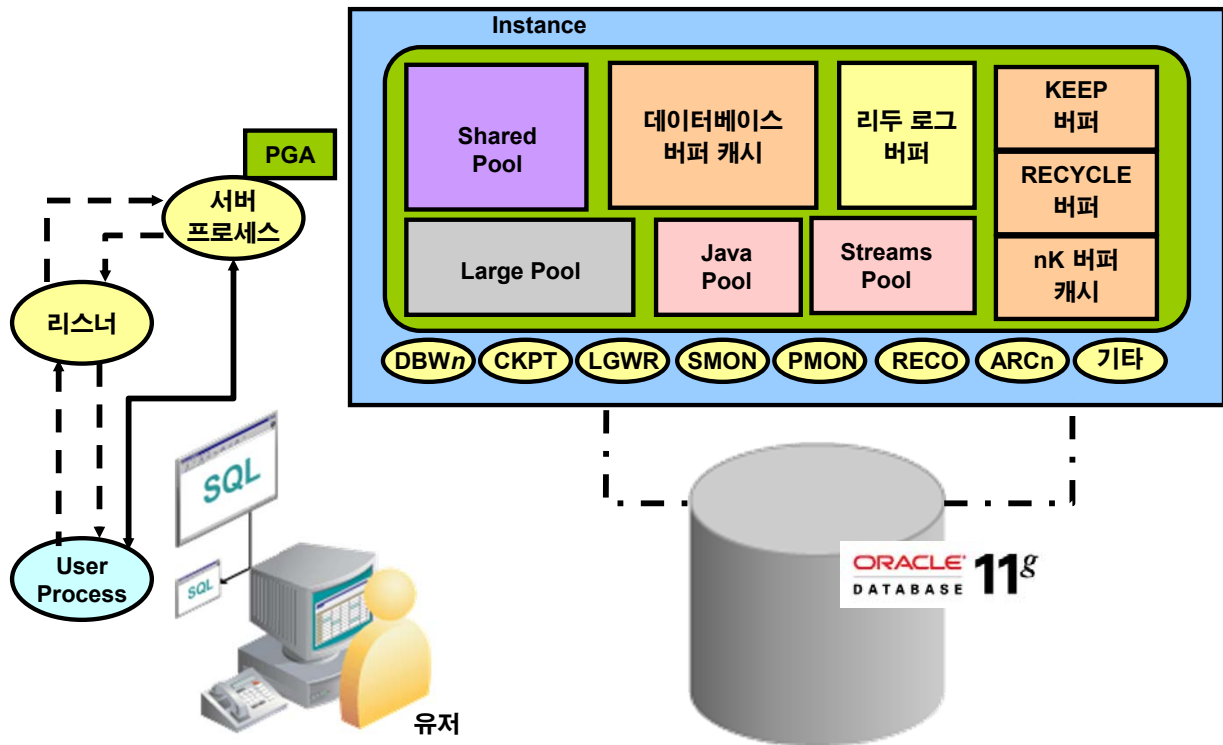
각 오라클 데이터베이스에는 SYSTEM 테이블스페이스 및 SYSAUX 테이블스페이스가 포함되어야 합니다. 이 테이블스페이스는 데이터베이스가 생성될 때 자동으로 생성됩니다. 시스템 기본값은 Small File 테이블스페이스를 생성하는 것입니다. Big File 테이블스페이스도 생성할 수 있는데 이 테이블스페이스를 사용하면 초대용량 파일을 관리할 수 있습니다.

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

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

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

오라클 데이터베이스와 상호 작용: 메모리, 프로세스 및 저장 영역



ORACLE

Copyright © 2010, Oracle. All rights reserved.

오라클 데이터베이스와 상호 작용

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

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

오라클 데이터베이스와 상호 작용(계속)

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

퀴즈

PMON(프로세스 모니터 프로세스):

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

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 2

퀴즈

ASM 파일은 어떤 유형의 Instance에서 액세스됩니까?

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

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 3

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

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

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

연습 1: 개요

이 연습은 다음에 대한 질문이 포함된 서면 연습입니다.

- 데이터베이스 구조
- 메모리
- 프로세스
- 파일 구조

ORACLE®

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 저장 영역 구조 관리

ORACLE

Copyright © 2010, Oracle. All rights reserved.

목표

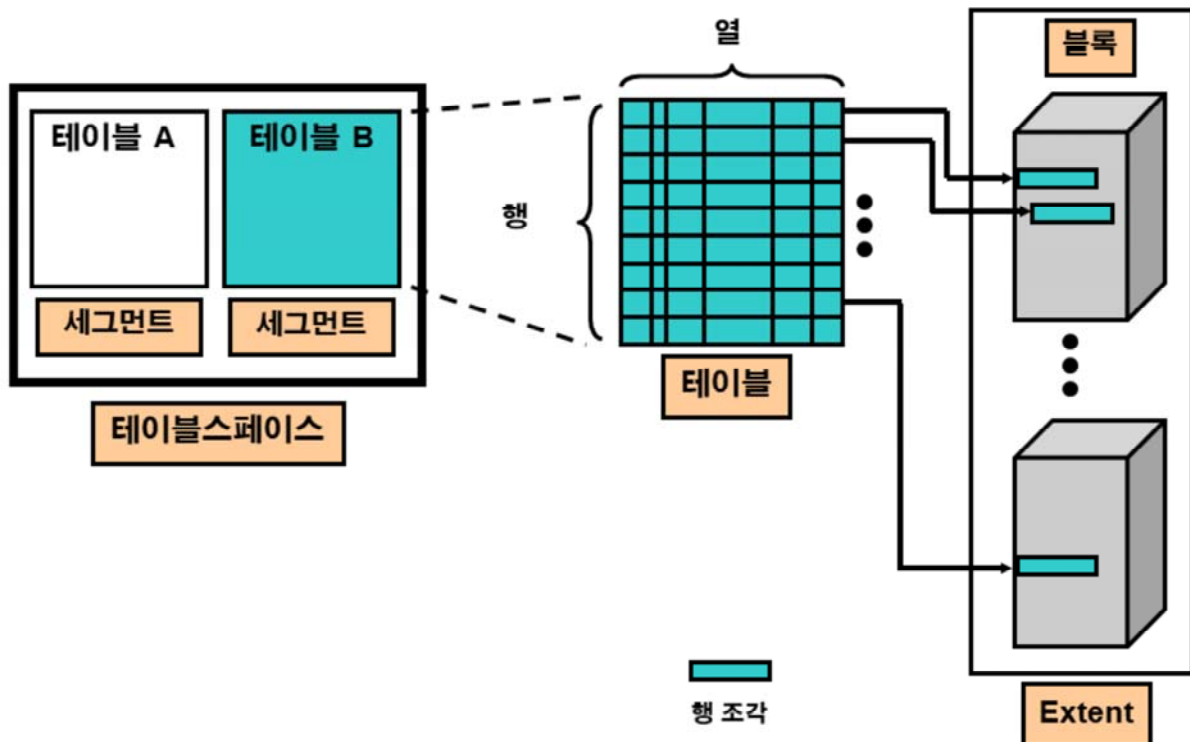
이 단원을 마치면 다음을 수행할 수 있습니다.

- 블록의 테이블 행 데이터 저장 영역 설명
- 테이블스페이스 생성 및 관리
- 테이블스페이스 정보 얻기

ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블 데이터가 저장되는 방법



ORACLE

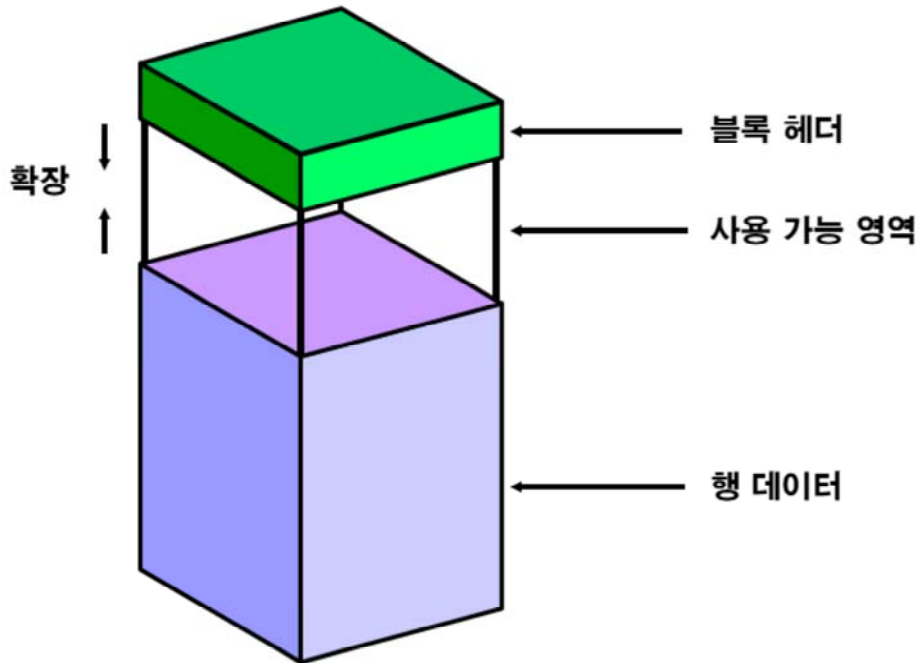
Copyright © 2010, Oracle. All rights reserved.

테이블 데이터가 저장되는 방법

테이블이 생성될 때 해당 테이블의 데이터를 보관하는 세그먼트가 생성됩니다. 테이블스페이스는 세그먼트 모음을 포함합니다.

논리적으로 테이블은 각 열의 값이 들어 있는 행들로 구성됩니다. 행은 행 조각의 품으로 결국 데이터베이스 블록에 저장됩니다. 행 조각이라고 하는 이유는 경우에 따라 전체 행이 한 장소에 저장되지 않을 수도 있기 때문입니다. 삽입된 행이 너무 커서 단일 블록(체인된 행)에 들어갈 수 없거나 기존 행이 갱신되어 현재 블록(이전된 행)의 사용 가능 영역보다 커지는 경우에 이러한 상황이 발생합니다. 행 조각은 또한 테이블의 열이 255개를 초과하는 경우에도 사용됩니다. 이런 경우에 조각들은 같은 블록(블록 내 체인화) 또는 다중 블록 간에 있기도 합니다.

데이터베이스 블록: 내용



ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 블록: 내용

- **블록 헤더:** 블록 헤더에는 세그먼트 유형(테이블이나 인덱스 등), 데이터 블록 주소, 테이블 디렉토리, 행 디렉토리, 블록의 행을 수정할 때 사용되는 각각 23바이트 정도의 트랜잭션 슬롯이 포함됩니다. 블록 헤더는 맨 위부터 아래쪽으로 확장됩니다.
- **행 데이터:** 블록에 있는 행의 실제 데이터입니다. 행 데이터 공간은 맨 아래부터 위쪽으로 확장됩니다.
- **사용 가능 영역:** 사용 가능 영역은 블록의 중간에 위치하며, 필요한 경우 헤더 및 행 데이터 공간이 늘어날 수 있도록 합니다. 새 행이 삽입되거나 기존 행의 열이 더 큰 값으로 갱신되면 행 데이터는 사용 가능 영역으로 확장됩니다. 헤더 확장을 유발하는 이벤트의 예는 다음과 같습니다.

- 행 항목이 추가로 필요한 행 디렉토리
- 처음에 구성한 것보다 많은 트랜잭션 슬롯 필요

처음에는 블록의 사용 가능 영역은 연속된 공간입니다. 그러나 삭제와 갱신을 거듭하면 블록의 사용 가능 영역이 단편화될 수 있습니다. 필요한 경우 블록의 사용 가능 영역은 Oracle 서버에 의해 병합됩니다.

저장 영역 구조 탐색



자세한 정보를 보려면 링크를
누르십시오.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

저장 영역 구조 탐색

논리적 데이터 구조는 데이터베이스의 물리적 파일에 저장됩니다. 데이터베이스의 논리적 구조는 EM(Enterprise Manager)을 통해 쉽게 확인할 수 있습니다. Server 페이지의 Storage 영역에 있는 링크를 누르면 각 구조에 대한 자세한 정보를 얻을 수 있습니다.

새 테이블스페이스 생성

Create Tablespace

Show SQL Cancel OK

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
Undo Retention Guarantee ☐ Yes ☒ No

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)
No items found		

ORACLE

Copyright © 2010, Oracle. All rights reserved.

새 테이블스페이스 생성

1. Server 탭을 누른 다음 Storage 머리글 아래에서 Tablespaces를 누릅니다.
2. Create를 누릅니다.
 주: 기존 테이블스페이스와 유사한 테이블스페이스를 생성하려면 기존 테이블스페이스를 선택하고 Actions 메뉴에서 Create Like를 선택합니다. Go를 누릅니다.
 Create Tablespace 페이지가 나타납니다.
3. 테이블스페이스의 이름을 입력합니다.
4. Extent Management 머리글 아래에서 Locally Managed를 선택합니다.
 로컬관리방식 테이블스페이스의 Extent는 오라클 데이터베이스 서버에 의해 테이블스페이스 내에서 효과적으로 관리됩니다. 딕셔너리 관리방식의 테이블스페이스의 경우에는 Extent를 더 능동적으로 관리해야 하며 Extent를 추적하기 위해 데이터 딕셔너리에 액세스해야 합니다. 딕셔너리 관리 방식의 옵션은 역호환성을 위해서만 제공되므로 사용하지 않을 것을 권장합니다.
5. Type 머리글 아래에서 Permanent를 선택합니다.
 영구 테이블스페이스에는 시스템 또는 사용자가 생성한 영구 데이터베이스 객체가 저장됩니다.
6. Status 머리글 아래에서 Read Write를 선택합니다.
 Read Write 상태는 테이블스페이스가 생성된 후 사용자가 테이블스페이스에 대한 읽기 및 쓰기 작업을 수행할 수 있음을 의미합니다. 이것은 기본값입니다.

7. 페이지의 Datafiles 영역에서 Add를 눌러 테이블스페이스에 데이터 파일을 추가합니다.

새 테이블스페이스 생성

Add Datafile (Cancel Continue)

Storage Type: Automatic Storage Management

* DiskGroup: DATA

Template: <Default>

Alias Directory:

Alias Name:

Tablespace: INVENTORY

File Size: 100 MB

☐ Reuse Existing File

Storage

☒ Automatically extend datafile when full (AUTOEXTEND)

Increment: 10 MB

Maximum File Size: ☒ Unlimited ☐ Value: MB

☒ TIP Changes made on this page will NOT take effect until you click the OK button.

Add Datafile

Storage Type: File System

* File Name:

* File Directory:

Tablespace: INVENTORY

File Size: 100 MB

☐ Reuse Existing File

적절한 저장 영역 유형 선택

ORACLE

Copyright © 2010, Oracle. All rights reserved.

새 테이블스페이스 생성(계속)

테이블스페이스에는 하나 이상의 파일이 있어야 합니다. 환경에 따라 적절한 저장 영역 유형을 선택합니다. Big File 테이블스페이스는 ASM 또는 기타 논리적 볼륨 관리자가 스트라이핑 또는 RAID(Redundant Array of Independent Disks) 및 동적으로 확장 가능한 논리적 볼륨을 지원하는 초대용량 데이터베이스에서 사용됩니다.

8. Add Datafiles 페이지에서 원하는 저장 영역 유형을 선택하고 필요한 정보를 입력합니다. ASM에서 원하는 디스크 그룹을 선택합니다. File System에 대해 데이터 파일에 대한 파일 이름 및 파일 디렉토리를 입력합니다.
9. 원하는 파일 크기를 입력합니다.
10. Storage 영역에서 "Automatically extend datafile when full (AUTOEXTEND)"를 선택한 다음 Increment 필드에 크기를 지정합니다. 이렇게 하면 공간이 부족할 때마다 데이터 파일이 자동으로 확장됩니다. 확장되는 공간은 물론 데이터 파일이 있는 물리적 매체의 크기에 의해 제한됩니다. Maximum File Size를 Unlimited로 지정하거나 최대 크기를 입력합니다. Continue를 눌러 Create Tablespace 페이지로 돌아갑니다.
12. Create Tablespace 페이지로 돌아가서 필요에 따라 Storage 탭을 눌러 이 테이블스페이스에 대한 저장 영역 옵션을 수정할 수 있습니다. 대부분의 경우 Storage 페이지에서 모든 기본값을 그대로 사용할 수 있습니다. OK를 눌러 테이블스페이스를 생성합니다.

주: 위 단계는 대부분의 상황에서 테이블스페이스를 빠르게 생성하는 방법을 보여줍니다. 구체적인 요구 사항에 따라 일부 옵션을 변경해야 할 수도 있습니다.

테이블스페이스에 대한 저장 영역

The screenshot shows the 'Storage' tab in the Oracle Database configuration window. It contains the following settings:

- Extent Allocation:**
 - ☒ Automatic
 - ☐ Uniform
 - Size: KB
- Segment Space Management:**
 - ☒ Automatic
Objects in the tablespace automatically manage their free space. It offers high performance for free space management.
 - ☐ Manual
Objects in the tablespace will manage their free space using free lists. It is provided for backward compatibility.
- Compression Options:**
 - Enabling data segment compression can reduce disk usage.
 - Compression: ☒ Disabled
 - ☐ Enabled on direct-path INSERT operations only
 - ☐ Enabled on all operations
- Enable logging:**
 - ☒ Yes
Generate redo logs for creation of tables, indexes and partitions, and for subsequent inserts. Recoverable
 - ☐ No
Redo log entries are smaller, the above operations are not logged and not recoverable.
- Block information:**
 - Block Size (B) 8192

ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블스페이스에 대한 저장 영역

Extent Allocation: 로컬관리방식 테이블스페이스 내의 Extent는 다음의 두 가지 방법 중 하나로 할당할 수 있습니다.

- **Automatic:** *autoallocate*라고도 하는 이 방법을 사용하면 테이블스페이스 내의 Extent 크기가 시스템에 의해 관리됩니다. 임시 테이블스페이스에 대해서는 Automatic을 지정할 수 없습니다.
- **Uniform:** 사용자가 지정한 동일 Extent 크기로 테이블스페이스가 관리됩니다. 기본 크기는 1MB입니다. 임시 테이블스페이스의 모든 Extent는 동일합니다. 언두 테이블스페이스에 대해서는 Uniform을 지정할 수 없습니다.

Segment Space Management: 로컬관리방식 테이블스페이스 내의 세그먼트 공간 관리는 다음과 같이 지정될 수 있습니다.

- **Automatic:** 오라클 데이터베이스는 비트맵을 사용하여 세그먼트 내의 사용 가능 영역을 관리합니다. 비트맵은 행 삽입에 사용할 수 있는 블록 내부 공간의 크기에 따라 세그먼트 내에 있는 각 데이터 블록의 상태를 설명합니다. 데이터 블록에서 사용 가능한 공간의 크기가 바뀌면 비트맵에 새로운 상태가 반영됩니다. 오라클 데이터베이스는 비트맵을 통해 사용 가능 영역을 보다 자동화된 방식으로 관리합니다. 따라서 이러한 공간 관리 유형을 Automatic Segment Space Management(ASMM)라고 합니다.

테이블스페이스에 대한 저장 영역(계속)

- **Manual:** 이 옵션을 지정하면 Free List를 사용하여 세그먼트 내의 사용 가능 영역을 관리할 수 있습니다. Free List는 행을 삽입할 공간이 있는 데이터 블록 리스트입니다. 이러한 형태의 세그먼트 공간 관리는 테이블스페이스에 생성된 스키마 객체에 대한 PCTUSED, FREELISTS 및 FREELIST GROUPS 저장 영역 파라미터를 지정 및 튜닝해야 하므로 *수동 세그먼트 공간 관리*라고 합니다. 이 기능은 역호환성을 위해 지원되는 기능이므로 대신 ASSM을 사용할 것을 권장합니다.

Compression Options: 데이터 세그먼트 압축은 기본적으로 비활성화되어 있습니다. 데이터 세그먼트 압축을 활성화하면 디스크 공간 사용을 저장하고, 버퍼 캐시의 메모리 사용을 줄이고, 읽기 작업 중 query 실행 속도를 높일 수 있습니다. 하지만 데이터 로딩 및 DML에 대한 CPU 오버헤드 비용이 듭니다. 이 방법은 특히 시간이 오래 걸리는 읽기 전용 작업이 있는 OLAP (online analytical processing) 시스템에 유용하지만 또한 OLTP(online transaction processing) 시스템에서도 사용될 수 있습니다.

압축 절을 사용하는 경우에 대한 자세한 내용은 *Oracle Database Administrator's Guide*를 참조하십시오.

Logging: Logging 절에서는 테이블스페이스에 생성된 세그먼트에 대한 기본 Logging 값을 설정합니다. 테이블스페이스의 객체에 대한 변경 내용은 리두 로그에 기록됩니다. Logging이 활성화되어 있지 않으면 SQL*Loader를 사용한 모든 직접 로드 및 직접 로드 INSERT 작업이 리두 로그에 기록되지 않으므로 데이터 손실이 발생할 경우 객체를 Recovery할 수 없게 됩니다. 따라서 Logging이 활성화되지 않은 상태에서 객체를 생성하는 경우 Recovery가 가능하도록 하려면 해당 객체를 백업해야 합니다. Logging 활성화를 선택하지 않으면 이후에 객체를 Recovery하는 기능에 상당한 영향을 줄 수 있습니다. 주의해서 사용하십시오. Logging 절에 대한 자세한

내용은 *Oracle Database SQL Reference*를 참조하십시오.

주: 데이터베이스가 FORCE LOGGING 모드 상태인 경우, 테이블스페이스 Logging 설정에 우선합니다. 데이터베이스는 데이터베이스 생성 시 또는 데이터베이스 생성 이후에 alter database force logging 명령을 사용하여 FORCE LOGGING 모드로 설정할 수 있습니다.

Block Information: 이 영역은 생성 중인 테이블스페이스에 사용되는 블록 크기를 보여줍니다. 여기에서는 이 크기가 읽기 전용 값으로 표시됩니다. 다른 블록 크기 초기화 파라미터(DB_nK_CACHE_SIZE)를 설정한 경우에는 해당 값이 여기에 옵션으로 나열됩니다. 다른 블록 크기를 정의하는 자세한 방법은 *Oracle Database Administrator's Guide*를 참조하십시오.

미리 구성된 데이터베이스의 테이블스페이스

- SYSTEM
- SYSAUX
- TEMP
- UNDOTBS1
- USERS
- EXAMPLE (선택 사항)

Tablespaces

Object Type: Tablespace

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name 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 Add Datafile Go

Select	Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/>	EXAMPLE	100.0	78.8	78.8	YES	21.2	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSAUX	697.2	663.9	95.2	YES	33.3	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSTEM	750.0	744.2	99.2	YES	5.8	✓	1	PERMANENT	LOCAL	MANUAL
<input type="radio"/>	TEMP	27.0	0.0	0.0	YES	27.0	✓	1	TEMPORARY	LOCAL	MANUAL
<input type="radio"/>	UNDOTBS1	100.0	16.1	16.1	YES	83.9	✓	1	UNDO	LOCAL	MANUAL
<input type="radio"/>	USERS	5.0	4.1	82.5	YES	0.9	✓	1	PERMANENT	LOCAL	AUTO

Total Allocated Size (GB) 1.64
Total Used (GB) 1.47
Total Allocated Free Space (GB) 0.17

✓ Online ✗ Offline ⚙ Read Only

ORACLE

Copyright © 2010, Oracle. All rights reserved.

미리 구성된 데이터베이스의 테이블스페이스

이 과정에서는 미리 구성된 데이터베이스에 다음 테이블스페이스가 생성됩니다.

- **SYSTEM:** SYSTEM 테이블스페이스는 Oracle 서버에서 데이터베이스 관리를 위해 사용합니다. 이 테이블스페이스에는 데이터베이스에 대한 관리 정보를 포함하는 테이블 및 데이터 덱서너리가 들어 있습니다. 이들은 모두 SYS 스키마에 포함되어 있으며 SYS 유저나 필요한 권한을 가진 관리 유저만 액세스할 수 있습니다.
- **SYSAUX:** SYSTEM 테이블스페이스의 보조 테이블스페이스입니다. 이전 오라클 데이터베이스 릴리스 중 일부 구성 요소 및 제품에서는 SYSTEM 테이블스페이스 또는 고유 테이블스페이스를 사용했지만 이제는 SYSAUX 테이블스페이스를 사용합니다. 모든 Oracle Database 10g Release 이상의 데이터베이스에는 SYSAUX 테이블스페이스가 있어야 합니다.

EM(Enterprise Manager)에서는 이 테이블스페이스의 내용을 파이 차트로 볼 수 있습니다. 파이 차트를 보려면 Administration 페이지에서 Tablespaces를 누릅니다. SYSAUX를 선택하고 Edit를 누릅니다. 그런 다음 Occupants 탭을 누릅니다. 생성한 후에는 EM을 사용하여 SYSAUX 테이블스페이스 내에 있는 각 보유자의 공간 사용량을 모니터할 수 있습니다. SYSAUX 테이블스페이스의 공간을 너무 많이 차지하는 구성 요소가 있거나 그럴 가능성이 있으면 보유자 중 하나를 선택하고 Change Tablespace를 눌러 보유자를 다른 테이블스페이스로 이동할 수 있습니다.

미리 구성된 데이터베이스의 테이블스페이스(계속)

- **TEMP:** 임시 테이블스페이스는 대규모 정렬이나 인덱스 생성 등과 같이 임시 세그먼트 생성이 필요한 SQL 문을 실행할 때 사용됩니다. 각 유저에게 생성된 데이터 객체를 저장하기 위한 기본 테이블스페이스가 할당되는 것과 마찬가지로 각 유저에게 임시 테이블스페이스가 할당됩니다. 가장 좋은 방법은 데이터베이스의 기본 임시 테이블스페이스를 정의하는 것입니다. 그러면 별도로 지정하지 않는 한 이 테이블스페이스가 모든 새로 생성된 유저에게 할당됩니다. 미리 구성된 데이터베이스에서는 TEMP 테이블스페이스가 기본 임시 테이블스페이스로 지정됩니다. 즉 유저 계정 생성 시 임시 테이블스페이스를 지정하지 않으면 오라클 데이터베이스는 이 테이블스페이스를 유저에게 할당합니다.
- **UNDOTBS1:** 데이터베이스 서버가 언두 정보를 저장하는 데 사용하는 언두 테이블스페이스입니다. 데이터베이스가 Automatic Undo Management를 사용하는 경우 지정된 시간에 단지 하나의 언두 테이블스페이스만 사용할 수 있습니다. 이 테이블스페이스는 데이터베이스가 생성될 때 생성됩니다.
- **USERS:** 이 테이블스페이스는 유저 객체 및 데이터를 저장하는 데 사용됩니다. 유저가 생성되었을 때 기본 테이블스페이스가 지정되지 않은 경우 USERS 테이블스페이스는 그 유저에 의해 생성된 모든 객체에 대한 기본 테이블스페이스입니다. SYS 및 SYSTEM 유저에게 기본 영구 테이블스페이스는 SYSTEM입니다.
- **EXAMPLE:** 이 테이블스페이스에는 데이터베이스 생성 시 설치할 수 있는 예제 스키마가 포함되어 있습니다. 이러한 예제 스키마는 예제에 사용할 수 있는 공통 플랫폼을 제공합니다. Oracle 설명서 및 교육용 소프트웨어에는 예제 스키마를 기반으로 하는 예제가 포함되어 있습니다.

주: 손쉬운 관리를 위해 별도의 인덱스용 테이블스페이스를 가지고 있는 것이 일반적입니다.

테이블스페이스 변경

Select Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
EXAMPLE	100.0	78.8	78.8	YES	21.2	✓	1	PERMANENT	LOCAL	AUTO
SYSAUX										

Edit Tablespace: EXAMPLE

Actions: Add Datafile Go Show SQL Revert Apply

General Storage Thresholds

Name: EXAMPLE

Bigfile tablespace: No

Extent Management

☒ Locally Managed
☐ Dictionary Managed

Type

☒ Permanent
☐ Set as default permanent tablespace
☐ Encryption (Encryption Options)
☐ Temporary
☐ Set as default temporary tablespace
☐ Undo

Status

☒ Read Write
☐ Read Only
☐ Offline
 Offline Mode: Normal

Datafiles

Edit Remove Add

Select Name	Directory	Size (MB)	Used (MB)
example_265_688820635	+DATA/orcl/datafile/	100.00	78.81

ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블스페이스 변경

테이블스페이스를 생성한 후 시스템 요구 사항이 변경되면 여러 가지 방법으로 테이블스페이스를 변경할 수 있습니다.

이름 변경: 테이블스페이스의 새 이름을 입력한 후 Apply를 누릅니다.

상태 변경: 테이블스페이스의 상태에는 세 가지가 있습니다. 테이블스페이스 유형에 따라 다음 세 가지 상태 중 일부를 사용하지 못할 수도 있습니다.

- **Read Write:** 테이블스페이스가 온라인이며 읽기 및 쓰기 작업을 수행할 수 있습니다.
- **Read Only:** Read Only를 지정하면 테이블스페이스가 변환 Read Only 모드로 설정됩니다. 이 상태에서 기존 트랜잭션을 완료(커밋 또는 롤백)할 수는 있지만 테이블스페이스의 객체에 대해 더 이상 DML(데이터 조작어) 작업을 수행할 수 없습니다. 테이블스페이스는 Read Only 상태에서 온라인입니다. SYSTEM 또는 SYSAUX 테이블스페이스를 읽기 전용으로 지정할 수는 없습니다.

주: 언두 및 임시 테이블스페이스는 읽기 전용으로 지정될 수 없습니다.

테이블스페이스 변경(계속)

- **Offline:** 온라인 테이블스페이스를 오프라인으로 설정하여 임시로 해당 부분의 데이터베이스를 일반적인 용도로 사용할 수 없도록 할 수 있습니다. 이 부분 이외의 데이터베이스는 열려 있어 사용자가 데이터에 액세스할 수 있습니다. 오프라인 상태로 설정할 때 다음 옵션을 사용할 수 있습니다.
 - **Normal:** 테이블스페이스의 모든 데이터 파일에 오류 조건이 없는 경우 정상적으로 테이블스페이스를 오프라인으로 설정할 수 있습니다. 오라클 데이터베이스는 테이블스페이스의 모든 데이터 파일을 오프라인으로 설정할 경우 해당 파일에 대해 체크포인트를 수행하는 방식으로 모든 데이터가 디스크에 기록되도록 합니다.
 - **Temporary:** 테이블스페이스의 하나 이상의 파일에 오류 조건이 있어도 임시로 테이블스페이스를 오프라인으로 설정할 수 있습니다. 오라클 데이터베이스는 아직 오프라인이 아닌 데이터 파일을 오프라인으로 설정하고 체크포인트를 수행합니다. 오프라인 파일이 없지만 임시 절을 사용하는 경우 테이블스페이스를 다시 온라인으로 되돌리는 데는 Media Recovery가 필요하지 않습니다. 단, 테이블스페이스에서 하나 이상의 파일이 쓰기 오류로 인해 오프라인이며 사용자가 테이블스페이스를 오프라인으로 임시 설정하는 경우 해당 테이블스페이스를 다시 온라인으로 되돌리려면 recovery가 필요합니다.
 - **Immediate:** 오라클 데이터베이스에서 데이터 파일의 체크포인트를 수행하지 않고 테이블스페이스를 즉시 오프라인으로 설정할 수 있습니다. Immediate를 지정한 경우 테이블스페이스를 온라인으로 되돌리려면 Media Recovery가 필요합니다. 데이터베이스가 NOARCHIVELOG 모드로 실행되고 있는 경우에는 테이블스페이스를 즉시 오프라인으로 설정할 수 없습니다.
 - **For Recover:** FOR RECOVER 설정은 지원이 중단되었습니다. 이 구문은 역호환성을 위해서만 지원됩니다.

주: 시스템 테이블스페이스는 오프라인으로 설정할 수 없습니다.

Changing the size: 테이블스페이스에 데이터 파일을 추가하거나 기존 데이터 파일의 크기를 변경하여 기존 테이블스페이스에 공간을 추가할 수 있습니다.

- 테이블스페이스에 새 데이터 파일을 추가하려면 Add를 누른 다음 Add Datafile 페이지에 데이터 파일에 대한 정보를 입력합니다.

주: Big File 테이블스페이스에 추가 데이터 파일을 추가할 수는 없습니다.

- 기존 데이터 파일의 크기를 변경하려면 Edit Tablespace 페이지의 Datafiles 영역에서 데이터 파일 이름을 눌러 데이터 파일을 선택하거나, 데이터 파일을 선택하고 Edit를 누릅니다. 그런 다음 Edit Datafile 페이지에서 데이터 파일의 크기를 변경할 수 있습니다. 테이블스페이스를 확장하거나 축소할 수 있지만 데이터 파일을 파일에서 사용된 공간보다 작게 만들 수 없습니다. 이렇게 하려고 시도하면 다음과 같은 오류 메시지가 표시됩니다.

```
ORA-03297: file contains used data beyond requested  
RESIZE value
```

Storage options: Storage를 눌러 테이블스페이스의 logging 방식을 변경합니다.

Thresholds: Thresholds를 눌러 테이블스페이스에서 warning 또는 critical 레벨의 공간 사용에 도달하는 시점을 변경합니다. 다음 세 가지 옵션이 있습니다.

- **Use Database Default Thresholds:** 사전 설정된 기본값을 사용합니다. 이 기본값을 설정하는 옵션이 제공됩니다.
- **Specify Thresholds:** 이 옵션을 선택하면 해당 테이블스페이스에 대한 임계값을 설정할 수 있습니다.
- **Disable Thresholds:** 이 옵션은 해당 테이블스페이스에 대한 공간 사용 Alert를 해제합니다.

주: 공간 활용률은 기본적으로 매 10분 마다 확인되며, 임계값 alert 등록에 몇 분이 걸릴 수도 있습니다.

테이블스페이스에서 가능한 작업

Selection Mode: Single

Buttons: Edit, View, Delete, Actions, Add Datafile, Go, Create

Select	Name	Allocated Size(MB)	Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/>	EXAMPLE	100.0	78.8	YES	21.2	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSAUX	697.2	95.2	YES	33.3	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSTEM	750.0	99.2	YES	5.8	✓	1	PERMANENT	LOCAL	MANUAL
<input type="radio"/>	TEMP	27.0	0.0	YES	27.0	✓	1	TEMPORARY	LOCAL	MANUAL
<input type="radio"/>	UNDOTBS1	100.0	17.1	YES	82.9	✓	1	UNDO	LOCAL	MANUAL
<input type="radio"/>	USERS	5.0	4.1	YES	0.9	✓	1	PERMANENT	LOCAL	AUTO

Total Allocated Size (GB) 1.64
Total Used (GB) 1.47
Total Allocated Free Space (GB) 0.17

Legend: ✓ Online, ✗ Offline, Ⓜ Read Only

Show DDL

```
CREATE SMALLFILE TABLESPACE "EXAMPLE" DATAFILE '+DATA/orcl/datafile/example.265.688820635'
SIZE 100M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M NOLOGGING EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블스페이스에서 가능한 작업

Actions 메뉴를 사용하여 테이블스페이스에서 다양한 작업을 수행할 수 있습니다.

테이블스페이스를 선택한 다음 수행할 작업을 선택합니다.

- **Add Datafile:** 테이블스페이스에 데이터 파일을 추가하여 테이블스페이스를 확장합니다.
- **Create Like:** 테이블스페이스를 템플릿으로 사용하여 다른 테이블스페이스를 생성합니다.
- **Generate DDL:** 테이블스페이스를 생성하는 DDL(데이터 정의어) 문을 생성합니다. 생성한 DDL 문을 복사하여 텍스트 파일에 붙여 넣은 다음 스크립트로 사용하거나 설명서에서 사용할 수 있습니다.
- **Make Locally Managed:** 테이블스페이스가 현재 디렉터리 관리 방식인 경우 로컬관리방식으로 변환합니다. 이 변환은 한쪽 방향으로만 가능합니다. 즉 테이블스페이스를 다시 디렉터리 관리 방식으로 변환할 수는 없습니다. 필요한 경우 PL/SQL 패키지인 DBMS_SPACE_ADMIN.TABLESPACE_MIGRATE_FROM_LOCAL을 사용하여 디렉터리 관리 방식으로 변환할 수도 있습니다.
- **Make Readonly:** 테이블스페이스에 대한 모든 쓰기 작업을 정지합니다. 현재 트랜잭션의 쓰기 작업은 완료할 수 있지만 테이블스페이스에서 새 DML 또는 다른 쓰기 작업을 시작할 수 없습니다. 테이블스페이스가 현재 읽기 전용이 아닌 경우에만 이 작업이 표시됩니다.
- **Make Writable:** 테이블스페이스의 객체에 DML 및 다른 쓰기 작업을 허용합니다. 테이블스페이스가 현재 쓰기 가능 상태가 아닌 경우에만 이 작업이 표시됩니다.

테이블스페이스에서 가능한 작업(계속)

- **Place Online:** 현재 오프라인 상태인 테이블스페이스를 온라인 상태로 만듭니다.
- **Reorganize:** Reorganization Wizard를 시작합니다. 이 마법사를 통해 테이블스페이스 내에서 객체를 이동하여 사용되지 않는 공간을 확보할 수 있습니다. 이 작업은 테이블스페이스 객체의 사용량이 적은 시간에 수행되어야 합니다.
- **Run Segment Advisor:** Segment Advisor를 시작합니다. Segment Advisor를 사용하면 객체 내의 공간 단편화 레벨을 기반으로 객체에 사용 가능한 공간이 있는지 확인할 수 있습니다. 테이블스페이스 레벨에서 테이블스페이스의 모든 세그먼트에 대해 권고가 생성됩니다.
- **Show Dependencies:** 해당 테이블스페이스가 종속되어 있는 객체나 해당 테이블스페이스에 종속된 객체를 표시합니다.
- **Show Tablespace Contents:** 모든 Extent의 그래픽 맵을 포함하여 테이블스페이스의 모든 세그먼트에 대한 정보를 보여줍니다.
- **Take Offline:** 현재 온라인 상태인 테이블스페이스를 사용할 수 없게 만듭니다. 테이블스페이스가 삭제되는 것은 아니며 단지 사용할 수 없는 상태가 됩니다.

테이블스페이스 삭제

Warning

Once a tablespace has been dropped, the objects and data in it will no longer be available. To recover them can be a time consuming process. Oracle recommends a backup before and after dropping a tablespace.

Are you sure you want to delete Tablespace EXAMPLE?

☒ Delete associated datafiles from storage

No Yes

Select	Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/>	EXAMPLE	100.0	78.8	78.8	YES	21.2	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSAUX	697.2	663.9	95.2	YES	33.3	✓	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSTEM	750.0	744.2	99.2	YES	5.8	✓	1	PERMANENT	LOCAL	MANUAL
<input type="radio"/>	TEMP	27.0	0.0	0.0	YES	27.0	✓	1	TEMPORARY	LOCAL	MANUAL
<input type="radio"/>	UNDOTBS1	100.0	17.1	17.1	YES	82.9	✓	1	UNDO	LOCAL	MANUAL
<input type="radio"/>	USERS	5.0	4.1	82.5	YES	0.9	✓	1	PERMANENT	LOCAL	AUTO

ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블스페이스 삭제

더 이상 필요하지 않은 테이블스페이스와 해당 콘텐츠(테이블스페이스에 포함된 세그먼트)를 데이터베이스에서 삭제할 수 있습니다. 테이블스페이스를 삭제하려면 DROP TABLESPACE 시스템 권한이 있어야 합니다.

테이블스페이스를 삭제하면 연관된 데이터베이스의 콘트롤 파일에 있는 파일 포인터가 제거됩니다. OMF(Oracle 관리 파일)를 사용하는 경우에는 기본 운영 체제 파일도 제거됩니다. OMF를 사용하지 않는 경우 삭제된 테이블스페이스를 구성하는 운영 체제 파일(데이터 파일)을 삭제하도록 지정할 수도 있습니다. 테이블스페이스를 삭제할 때 Oracle 서버에서 데이터 파일도 함께 삭제하도록 지정하지 않은 경우 나중에 데이터 파일을 삭제하려면 운영 체제의 해당 명령을 사용해야 합니다.

활성 세그먼트가 포함된 테이블스페이스는 삭제할 수 없습니다. 예를 들어, 테이블스페이스에 있는 테이블이 현재 사용 중이거나 테이블스페이스에 커밋되지 않은 트랜잭션을 롤백하는 데 필요한 언두 데이터가 포함되어 있는 경우에는 해당 테이블스페이스를 삭제할 수 없습니다. 테이블스페이스는 온라인이어도 되고 오프라인이어도 되지만 삭제하기 전에 오프라인으로 설정하는 것이 좋습니다.

테이블스페이스 정보 보기

```
SELECT tablespace_name, status, contents, logging, extent_management,
allocation_type, segment_space_management
FROM dba_tablespaces
```

TABLESPACE_NAME	STATUS	CONTENTS	LOGGING	EXTENT_MANAGEMENT	ALLOCATION_TYPE	SEGMENT_SPACE_MANAGEMENT
SYSTEM	ONLINE	PERMANENT	LOGGING	LOCAL	SYSTEM	MANUAL
SYSAUX	ONLINE	PERMANENT	LOGGING	LOCAL	SYSTEM	AUTO
UNDOTBS1	ONLINE	UNDO	LOGGING	LOCAL	SYSTEM	MANUAL
TEMP	ONLINE	TEMPORARY	NOLOGGING	LOCAL	UNIFORM	MANUAL
USERS	ONLINE	PERMANENT	LOGGING	LOCAL	SYSTEM	AUTO
EXAMPLE	ONLINE	PERMANENT	NOLOGGING	LOCAL	SYSTEM	AUTO

```
SELECT ts#, name FROM v$tablespace
```

TS#	NAME
0	SYSTEM
1	SYSAUX
2	UNDOTBS1
4	USERS
3	TEMP
6	EXAMPLE

ORACLE

Copyright © 2010, Oracle. All rights reserved.

테이블스페이스 정보 보기

선택된 테이블스페이스에 대한 정보를 보려면 View를 누릅니다. View Tablespace 페이지에서 Edit를 눌러 테이블스페이스를 변경할 수도 있습니다.

다음은 query하여 테이블스페이스 및 데이터 파일에 대한 정보를 얻을 수도 있습니다.

- 테이블스페이스 정보:
 - DBA_TABLESPACES
 - V\$TABLESPACE
- 데이터 파일 정보:
 - DBA_DATA_FILES
 - V\$DATAFILE

주: V\$DBFILE 뷰에는 데이터베이스의 모든 데이터 파일이 표시됩니다. 이 뷰는 기록 호환성을 위해 보존됩니다. 대신 V\$DATAFILE을 사용하는 것이 권장됩니다.

- 임시 파일 정보:
 - DBA_TEMP_FILES
 - V\$TEMPFILE

테이블스페이스 내용 보기

Size (MB)100.0

Block Size (KB)8

Used (MB)78.8

Used (%)78.8

Extent Mgmt

Segment Mgmt

LOCAL

AUTO

Auto Extend

Extents

Yes

882

Return

Segments

Search

Segment Name

Type

Minimum Size

Minimum Extents

All Types

(KB)

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.

Segment Name

SH.CUSTOMERS

PM.SYS_LOB0000

PM.SYS_LOB0000

SH.SUPPLEMENTA

OEPRODUCT_DES

SH.SALES.SALES_

SH.SALES.SALES_

SH.SALES.SALES_

SH.SALES.SALES_

SH.CUSTOMERS_P

SH.SALES.SALES_

Extent Map

Clicking the Highlight Extents button for a segment in the table will cause all extents that belong to that segment to be highlighted in the Extent Map. Clicking on a used extent in the Extent Map will select the segment to which that extent belongs in the segment table.

Header

Used

Free

Selected

Unmapped

Zoom

75%

Extent Map

ORACLE

Copyright © 2010, Oracle. All rights reserved.

기본 테이블스페이스 페이지 또는 특정 테이블스페이스 페이지에서, Actions drop-down list에서 Show Tablespace Contents를 선택한 다음 Go를 누릅니다. Show Tablespace Contents 페이지에는 테이블스페이스의 세그먼트 리스트, 각 세그먼트의 유형, 세그먼트 크기 및 각 세그먼트의 Extent 개수 등 테이블스페이스에 대한 상세한 정보가 표시됩니다. 열 헤더를 눌러 리스트를 정렬하거나 Search 영역에 값을 입력하여 리스트를 필터링할 때 이 네 가지 값을 사용할 수 있습니다. 디렉터리 관리방식의 테이블스페이스인 경우 다음과 같은 추가 열이 표시됩니다.

- Extent list를 보려면 Extents 열의 링크를 누릅니다.

- Extent가 속한 세그먼트 이름
- Extent ID
- 블록 ID
- Extent 크기(블록 단위)
- Extent가 저장된 데이터 파일

OMF(Oracle 관리 파일)

파일 이름이 아닌 데이터베이스 객체 관점에서 파일 작업을 지정합니다.

파라미터	설명
DB_CREATE_FILE_DEST	데이터 파일 및 임시 파일에 대한 기본 파일 시스템 디렉토리의 위치를 정의합니다.
DB_CREATE_ONLINE_LOG_DEST_n	리두 로그 파일 및 콘트롤 파일 생성을 위한 위치를 정의합니다.
DB_RECOVERY_FILE_DEST	Fast Recovery Area의 기본 위치입니다.

예제:

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '+DATA';  
SQL> CREATE TABLESPACE tbs_1;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

OMF(Oracle 관리 파일)

Oracle 관리 파일을 사용하면 오라클 데이터베이스를 구성하는 운영 체제 파일을 직접 관리하지 않아도 됩니다. 파일 이름이 아닌 데이터베이스 객체 관점에서 작업을 지정합니다. 그러면 데이터베이스는 내부적으로 표준 파일 시스템 인터페이스를 사용하여 다음 데이터베이스 구조에 대한 파일을 필요에 따라 생성하고 삭제합니다.

- 테이블스페이스
- 리두 로그 파일
- 콘트롤 파일
- 아카이브된 로그
- 블록 변경 사항 추적 파일
- Flashback 로그
- RMAN 백업

데이터베이스에는 Oracle 관리 파일과 비관리 파일이 섞여 있을 수 있습니다. 데이터베이스에서 디렉토리를 직접 생성하지 않으므로 이러한 파라미터 중 하나로 지정된 파일 시스템 디렉토리가 존재해야 합니다. 또한 디렉토리에는 데이터베이스에서 해당 디렉토리에 파일을 생성하도록 허용하는 권한이 있어야 합니다.

위의 예제는 DB_CREATE_FILE_DEST가 설정된 후에는 CREATE TABLESPACE 문에서 DATAFILE 절이 생략될 수 있음을 보여줍니다. 데이터 파일은 DB_CREATE_FILE_DEST에 의해 지정된 위치에 생성됩니다. 예제에서처럼 테이블스페이스를 생성하면 모든 파라미터에

기본값이 할당됩니다.

OMF(Oracle 관리 파일)(계속)

Oracle 관리 파일은 고유의 이름 지정 형식이 있습니다. 예를 들어, Linux 및 Unix 기반 시스템에서는 다음 형식을 사용합니다.

```
<destination_prefix>/ol_mf_%t_%u_.dbf
```

Oracle 관리 파일 이름은 변경하지 마십시오. 데이터베이스는 Oracle 관리 파일을 이름으로 식별합니다. 파일 이름을 변경하는 경우 데이터베이스는 더 이상 Oracle 관리 파일로 인식할 수 없으며, 파일을 올바르게 관리할 수 없습니다.

다음 예제는 데이터 파일 생성 위치를 /u01/oradata로 설정하고 나서 이 위치에 데이터 파일이 포함된 tbs_1 테이블스페이스를 생성합니다.

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/u01/oradata';  
SQL> CREATE TABLESPACE tbs_1;
```

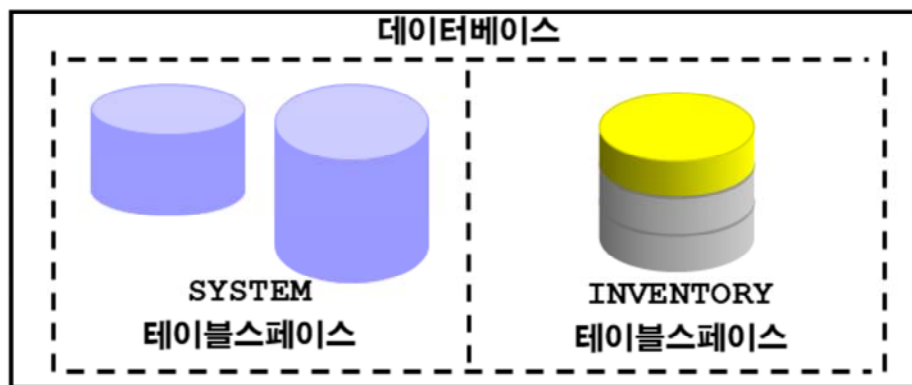
기본적으로 SYSTEM 및 SYSAUX 테이블스페이스를 포함한 Oracle 관리 데이터 파일은 100MB이며 자동 확장됩니다.

주: 기본적으로 ASM은 OMF 파일을 사용하지만, 테이블스페이스 생성 시 또는 기존 테이블스페이스에 ASM 데이터 파일을 추가할 때 ASM 데이터 파일에 대한 alias 이름을 지정한 경우에는 해당 파일이 OMF가 될 수 없습니다.

데이터베이스 확장

다음과 같은 방법으로 데이터베이스를 확장할 수 있습니다.

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



ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 확장

Enterprise Manager 또는 SQL 문을 사용하여 이 작업을 수행할 수 있습니다. 데이터베이스의 크기는 모든 테이블스페이스 크기의 합계로 설명될 수 있습니다.

퀴즈

데이터베이스에는 Oracle 관리 파일과 비관리 파일이 섞여 있을 수 있습니다.

1. 참
2. 거짓

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 1

퀴즈

Big File 테이블스페이스에는 적어도 100MB의 파일이
한 개 있어야 합니다.

1. 참
2. 거짓

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 2

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 블록의 테이블 행 데이터 저장 영역 설명
- 테이블스페이스 생성 및 관리
- 테이블스페이스 정보 얻기

ORACLE

Copyright © 2010, Oracle. All rights reserved.

연습 7 개요: 데이터베이스 저장 영역 구조 관리

이 연습에서는 다음 내용을 다룹니다.

- 테이블스페이스 생성
- 테이블스페이스에 대한 정보 수집

ORACLE

Copyright © 2010, Oracle. All rights reserved.

3 유저 보안 관리

ORACLE

Copyright © 2010, Oracle. All rights reserved.

목표

이 단원을 마치면 다음을 수행할 수 있습니다.

- **데이터베이스 유저 계정 생성 및 관리:**
 - 유저 인증
 - 기본 저장 영역(테이블스페이스) 할당
- **권한 부여 및 취소**
- **롤 생성 및 관리**
- **프로파일 생성 및 관리:**
 - 표준 암호 보안 기능 구현
 - 유저별 리소스 사용량 제어

ORACLE

Copyright © 2010, Oracle. All rights reserved.

목표

다음은 데이터베이스 유저를 관리하고 목표를 이해하는 데 도움이 되는 관련 용어입니다.

- *데이터베이스 유저 계정(Database User Account)*은 데이터베이스 객체에 대한 소유권 및 액세스 권한을 구성하는 수단입니다.
- *암호(password)*는 오라클 데이터베이스에 의한 인증입니다.
- *권한(privilege)*은 특정 유형의 SQL 문을 실행하거나 다른 유저의 객체에 액세스할 수 있는 권한입니다.
- *롤(Role)*은 유저나 다른 롤에 부여된 관련 권한의 명명된 그룹입니다.
- *프로파일(Profile)*은 데이터베이스 사용 및 Instance 리소스에 명명된 리소스 제한 집합을 적용하고 계정 상태 및 암호 관리 규칙을 관리합니다.
- *할당량(Quota)*은 주어진 테이블스페이스에 허용된 공간입니다. 할당량은 유저의 리소스 사용을 제어할 수 있는 방법 중 하나입니다.

데이터베이스 유저 계정

각 데이터베이스 유저 계정에는 다음이 포함되어 있습니다.

- 고유 Username
- 인증 방식
- 기본 테이블스페이스
- 임시 테이블스페이스
- 유저 프로필
- 초기 Consumer Group
- 계정 상태



스키마:

- 데이터베이스 유저가 소유하는 데이터베이스 객체의 모음
- 유저 계정과 동일한 이름을 가짐

ORACLE

Copyright © 2010, Oracle. All rights reserved.

데이터베이스 유저 계정

데이터베이스에 액세스하려면 유저가 유효한 데이터베이스 유저 계정을 지정하고 이 유저 계정에 필요한 인증에 성공해야 합니다. 각 데이터베이스 유저에게는 고유한 데이터베이스 계정이 있습니다. 오라클은 잠재적인 보안 취약성을 방지하고 특정 감사(audit) 작업에 대해 의미 있는 데이터를 제공하기 위해 고유한 데이터베이스 계정을 사용할 것을 권장합니다. 그러나 유저가 일반 데이터베이스 계정을 공유하는 경우가 있습니다. 이 경우에 운영 체제와 응용 프로그램은 데이터베이스에 대해 적절한 보안을 제공해야 합니다. 각 유저 계정에는 다음이 포함됩니다.

- **고유 Username:** Username은 30바이트를 초과하거나 특수 문자를 포함할 수 없으며 문자로 시작되어야 합니다.
- **인증 방식:** 가장 일반적인 인증 방식은 암호이지만, Oracle Database 11g에서는 Password, Global 및 External 인증 방식(예: 생물 통계학적(Biometrics) 인증, 인증서 및 토큰 인증)을 지원합니다.
- **기본 테이블스페이스:** 유저가 다른 테이블스페이스를 지정하지 않은 경우 객체를 생성하는 장소입니다. 기본 테이블스페이스가 있다고 해서 유저에게 이 테이블스페이스에 객체를 생성할 수 있는 권한이 있거나 기본 테이블스페이스 내에 객체를 생성할 공간 할당량이 있는 것은 아니라는 점에 유의하십시오. 이는 모두 별도로 부여됩니다.
- **임시 테이블스페이스:** Instance가 정렬 및 임시 테이블 등의 임시 객체를 유저 대신 생성하는 위치입니다. 임시 테이블스페이스에는 할당량이 적용되지 않습니다.

데이터베이스 유저 계정(계속)

- **유저 프로파일:** 유저에게 할당되는 리소스 및 암호 제한 집합입니다.
- **초기 Consumer Group:** 리소스 관리자가 사용합니다.
- **계정 상태:** 유저는 "열린" 계정에만 액세스할 수 있습니다. `account_status`는 "locked" 상태와 "expired" 상태가 다양하게 조합된 형태일 수 있습니다.

스키마: 스키마는 데이터베이스 유저가 소유하는 데이터베이스 객체의 모음입니다. 스키마 객체는 데이터베이스의 데이터를 직접 참조하는 논리적 구조입니다. 스키마 객체에는 테이블, 뷰, 시퀀스, 내장 프로시저, 동의어, 인덱스, 클러스터 및 데이터베이스 링크가 있습니다. 일반적으로 스키마 객체에는 응용 프로그램이 데이터베이스에 생성하는 모든 것이 포함됩니다.

주: 데이터베이스 유저가 반드시 사람일 필요는 없습니다. 즉, 일반적으로 HR과 같은 특정 응용 프로그램의 데이터베이스 객체를 소유하는 유저를 생성하게 됩니다. 데이터베이스 유저는 장치나 응용 프로그램일 수도 있고 보안용으로 데이터베이스 객체를 그룹화하는 방식일 수도 있습니다. 데이터베이스 유저에게는 특정인의 개인 식별 정보가 필요하지 않습니다.

미리 정의된 관리 계정

- **SYS 계정:**
 - DBA 롤 및 기타 몇 가지 롤이 부여됨
 - ADMIN OPTION과 함께 모든 권한을 가짐
 - 시작, 종료 및 일부 유지 관리 명령에 필요
 - 데이터 디렉터리 및 AWR(Automatic Workload Repository) 소유
- **SYSTEM** 계정은 DBA, MGMT_USER 및 AQ_ADMINISTRATOR_ROLE 롤이 부여됨
- **DBSNMP** 계정은 OEM_MONITOR 롤이 부여됨
- **SYSMAN** 계정은 MGMT_USER, RESOURCE 및 SELECT_CATALOG_ROLE 롤이 부여됨
- 이 계정들은 일상적인 작업에는 사용되지 않습니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

미리 정의된 관리 계정

SYS 및 SYSTEM 계정에는 기본적으로 데이터베이스 관리자(DBA) 롤이 부여됩니다. 또한 SYS 계정은 ADMIN OPTION에 대한 모든 권한을 포함하며 데이터 디렉터리를 소유합니다. SYS 계정에 연결하려면 데이터베이스 Instance에 대해 AS SYSDBA 절을 사용해야 하며 자동 저장 영역 관리(ASM) Instance에 대해서는 AS SYSASM을 사용해야 합니다. SYSDBA 권한이 부여된 모든 유저는 AS SYSDBA 절을 사용하여 SYS 계정에 연결할 수 있습니다. SYSDBA, SYSOPER 또는 SYSASM 권한이 부여된 "권한 있는" 유저만이 Instance를 시작 및 종료할 수 있습니다. SYSTEM 계정에는 SYSDBA 권한이 없습니다. 또한 SYSTEM에는 AQ_ADMINISTRATOR_ROLE 및 MGMT_USER 롤이 부여됩니다. SYS 및 SYSTEM 계정은 데이터베이스에서 필수 계정이며 삭제할 수 없습니다.

DBSNMP 계정은 Enterprise Manager의 관리 에이전트가 데이터베이스를 모니터 및 관리하는 데 사용됩니다. SYSMAN 계정은 Oracle Enterprise Manager 관리 작업을 수행하는 데 사용됩니다. DBSNMP 또는 SYSMAN은 SYSDBA 권한이 없습니다.

최적의 사용법(Best Practice Tip): 이러한 계정은 최소 권한 원칙이 적용되며 일상적인 작업에는 사용되지 않습니다. DBA 권한이 필요한 유저는 부여된 필수 권한과 함께 별도의 권한을 가집니다. 예를 들어, Jim에게는 jim이라는 낮은 권한의 계정과 jim_dba라는 높은 권한의 계정이 부여됩니다. 이 방법을 사용하면 최소 권한 원칙을 적용할 수 있고 계정을 공유할 필요가 없으며 개별 작업을 감사(audit)할 수 있습니다.

유저 생성

Database Instance: orcl.oracle.com > Users > Create User

Logged in As SYS

Show SQL Cancel OK

General Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

* Name: mydba

Profile: DEFAULT

Authentication: Password

* Enter Password: *****

* Confirm Password: *****

For Password choice, the role is authorized via password.

☐ Expire Password now

Default Tablespace: USERS

Temporary Tablespace: TEMP

Status: ☐ Locked ☒ Unlocked

Show SQL

Return

```
CREATE USER "MYDBA" PROFILE "DEFAULT" IDENTIFIED BY "*****" DEFAULT
TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK
GRANT "CONNECT" TO "MYDBA"
```

Server > Users를 선택한 다음 Create 버튼을 누릅니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

유저 생성

Enterprise Manager의 Users 페이지에서는 현재 데이터베이스에 액세스하도록 허용되는 데이터베이스 유저를 관리할 수 있습니다. 이 페이지를 사용하여 유저를 생성, 삭제하고 설정을 수정할 수 있습니다.

데이터베이스 유저를 생성하려면 다음을 수행하십시오.

1. Enterprise Manager Database Control에서 Server 탭을 누른 후에 Security 섹션에서 Users를 누릅니다.
2. Create 버튼을 누릅니다.

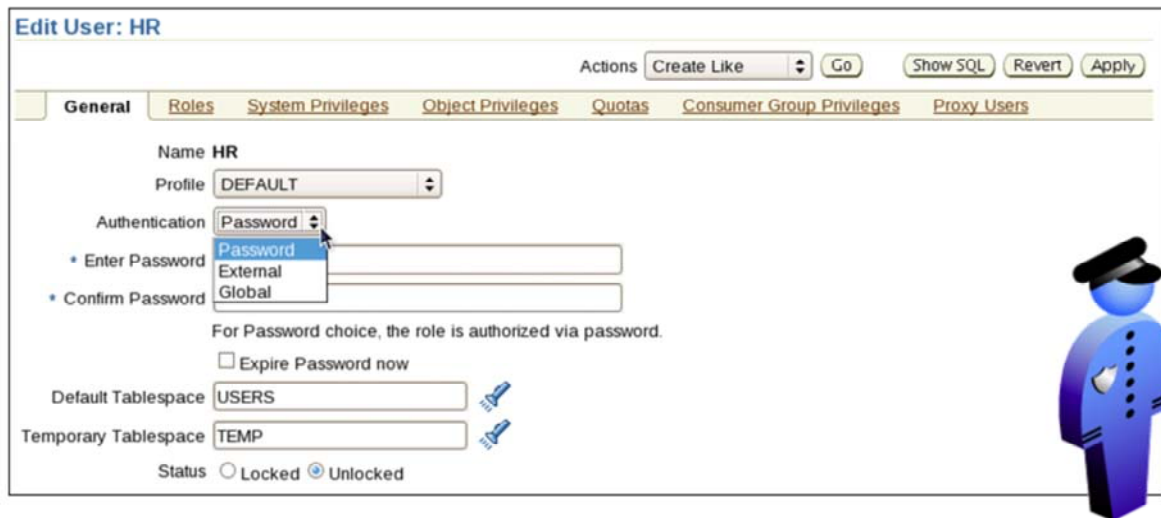
필요한 정보를 입력합니다. Name과 같은 필수 항목에는 별표(*)가 표시되어 있습니다. 지정된 이름은 데이터베이스 객체를 생성하는 데 사용되는 것과 동일한 규칙을 따라야 합니다. 이 단원의 다음 페이지에서는 인증에 대한 자세한 정보를 제공합니다. 프로파일은 이 단원 뒷부분에서 다룹니다.

각 유저에게 기본 테이블스페이스 및 임시 테이블스페이스를 할당합니다. 객체를 생성할 때 유저가 테이블스페이스를 지정하지 않으면 객체는 객체 소유자에게 할당된 기본 테이블스페이스에 생성됩니다. 그러므로 객체가 생성되는 위치를 제어할 수 있습니다. 기본 테이블스페이스를 선택하지 않으면 시스템에 정의된 기본 영구 테이블스페이스가 사용됩니다. 이는 임시 테이블스페이스에도 마찬가지인데, 테이블스페이스를 지정하지 않으면 시스템에 정의된 임시 테이블스페이스가 사용됩니다.

주: 지원되는 SQL 구문을 보려면 Show SQL을 누릅니다. 유저 생성을 위한 전체 SQL 구문은 Oracle® Database SQL Language Reference 설명서를 참조하십시오.

유저 인증

- Password
- External
- Global



ORACLE

Copyright © 2010, Oracle. All rights reserved.

유저 인증

인증은 데이터, 리소스 또는 응용 프로그램을 사용하려는 주체(유저, 장치 또는 기타 엔티티)의 ID를 검증하는 작업을 의미합니다. ID를 검증하면 원활한 상호 작용을 위한 신뢰 관계가 수립됩니다. 또한 인증을 통해 액세스 권한 및 작업을 특정 ID와 연결할 수 있으므로 책임 소재를 명백히 구분할 수 있습니다. 인증 이후에는 권한 부여 프로세스를 통해 특정 엔티티에 허용된 작업 및 액세스 권한 레벨을 허용하거나 제한할 수 있습니다.

유저를 생성할 때는 이 유저에 사용할 인증 기법을 결정해야 합니다. 이때 결정한 인증 기법은 나중에 수정할 수 있습니다.

Password: 오라클 데이터베이스에 의한 인증을 의미합니다. 유저가 연결을 시도할 때 입력해야 하는 연관된 암호와 함께 각 유저를 생성합니다. 암호 설정 시에는 해당 암호가 즉시 만료되도록 할 수 있으며 이렇게 하면 유저는 첫 로그인 후에 암호를 변경해야 합니다. 유저 암호를 만료시키려면 먼저 유저가 암호를 변경할 수 있는지 여부를 확인합니다. 일부 응용 프로그램에는 이러한 기능이 없습니다. Oracle Database 11g에서 생성한 모든 암호는 기본적으로 대소문자를 구분하며, 멀티바이트 문자를 포함할 수 있고 30바이트로 제한됩니다. Oracle Database 11g로 업그레이드된 데이터베이스에서 생성한 각 암호는 변경될 때까지 대소문자를 구분하지 않습니다. 암호는 네트워크를 통해 전송하기 전에 네트워크(클라이언트/서버 및 서버/서버) 연결 중에 AES(Advanced Encryption Standard) 알고리즘을 사용하여 항상 자동으로 투명하게 암호화(encryption)됩니다.

유저 인증(계속)

External: 데이터베이스 외부의 방법(운영 체제, Kerberos 또는 Radius)을 사용하는 인증입니다. Kerberos 또는 Radius의 경우에는 Advanced Security Option이 필요합니다. Username이나 암호를 지정하지 않고 오라클 데이터베이스에 쉽게 연결할 수 있습니다. 강력한 인증 방법인 Advanced Security Option을 사용하면 생물 통계학적(Biometrics) 인증, x509 인증 및 토큰 장치를 사용하여 유저를 식별할 수 있습니다. External 인증을 사용하면 데이터베이스는 기본 운영 체제, 네트워크 인증 서비스 또는 외부 인증 서비스를 사용하여 데이터베이스 계정에 대한 액세스를 제한합니다. 데이터베이스 암호는 이러한 로그인 유형에 사용되지 않습니다. 운영 체제 또는 네트워크 서비스에서 허용하는 경우에는 데이터베이스 암호를 사용하여 유저를 인증할 수 있습니다. 운영 체제 인증을 사용하려면 OS_AUTHENT_PREFIX 초기화 파라미터를 설정하고 이 접두어를 Oracle Username에 사용합니다. OS_AUTHENT_PREFIX 파라미터는 오라클 데이터베이스가 각 유저의 운영 체제 계정 이름 앞에 추가하는 접두어를 정의합니다. Oracle 소프트웨어의 역호환성을 위해 이 파라미터의 기본값은 OPS\$입니다. 유저가 연결을 시도할 때 오라클 데이터베이스는 접두어가 있는 Username을 데이터베이스에 있는 Oracle Username과 비교합니다. 예를 들어, OS_AUTHENT_PREFIX가 아래와 같이 설정되었다고 가정합니다.

OS_AUTHENT_PREFIX=OPS\$

운영 체제 계정 이름이 tsmith인 유저가 오라클 데이터베이스에 연결하여 운영 체제에 의해 인증 받는 경우 오라클 데이터베이스는 해당 데이터베이스 유저인 OPS\$tsmith가 있는지 확인하고 이 유저가 있는 경우 유저의 연결을 허용합니다. 운영 체제에서 인증되는 유저에 대한 모든 참조는 OPS\$tsmith와 같이 접두어를 포함해야 합니다.

주: 일부 운영 체제에서는 OS_AUTHENT_PREFIX 초기화 파라미터 텍스트의 대소문자를 구분합니다. 이 초기화 파라미터에 대한 자세한 내용을 보려면 사용 중인 운영 체제와 관련된 오라클 설명서를 참조하십시오.

Global: Oracle Advanced Security Option으로 Global 인증을 사용하면 Oracle Internet Directory를 사용하여 유저를 식별할 수 있습니다.

고급 인증 방식에 대한 자세한 사항은 *Oracle Database Security* 과정을 참조하십시오.

관리자 인증

운영 체제 보안:

- DBA는 파일을 생성하고 삭제하기 위한 OS 권한을 가져야 합니다.
- 일반 데이터베이스 유저는 데이터베이스 파일을 생성 또는 삭제할 수 있는 OS 권한을 가질 수 없습니다.

관리자 보안:

- SYSDBA, SYSOPER 및 SYSASM 연결:
 - Password file 및 강력한 인증 방식에 대해 DBA 유저가 이름별로 감사(audit)됩니다.
 - OS 인증에 대해 OS 계정 이름이 감사됩니다.
 - 권한이 있는 유저에 대해서는 OS 인증이 Password file 인증보다 우선합니다.
 - Password file은 대소문자를 구분하는 암호를 사용합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

관리자 인증

운영 체제 보안: UNIX 및 Linux에서 DBA는 기본적으로 데이터베이스 파일을 생성 및 삭제하는데 필요한 권한을 가진 oinstall OS 그룹에 속합니다.

관리자 보안: 권한이 있는 SYSDBA, SYSOPER 및 SYSASM 유저에 대한 연결은 Password file 또는 OS 권한을 사용한 검증을 수행한 이후에만 인증됩니다. OS 인증이 사용되는 경우 데이터베이스는 제공되는 Username과 암호를 사용하지 않습니다. OS 인증은 Password file이 없거나, 제공된 Username 또는 암호가 해당 파일에 없거나, Username과 암호가 제공되지 않는 경우에 사용됩니다. Oracle Database 11g의 Password file은 기본적으로 대소문자를 구분하는 암호를 사용합니다.

그러나 Password file을 통한 인증에 성공하면 해당 Username으로 연결이 기록됩니다.

운영 체제를 통한 인증에 성공하면 특정 유저를 기록하지 않는 CONNECT / 연결이 적용됩니다.

주: 운영 체제의 OSDBA 또는 OSOPER 그룹 멤버이고, SYSDBA 또는 SYSOPER로 연결하는 경우에는 지정한 Username 및 암호에 관계없이 관련된 관리 권한으로 연결됩니다. SYSASM의 경우 Username 또는 암호(예: sqlplus / as SYSASM)를 지정하지 않아야 합니다.

Oracle Database 11g에서 권한이 있는 유저는 Advanced Security Option에 대한 사용권이 있는 경우 Kerberos, SSL 또는 디렉토리 인증 등의 강력한 인증 방식을 사용할 수 있습니다.

유저 계정 Lock 해제 및 암호 재설정

Users

Object Type: User

Search

Enter an object name to filter the data that is displayed in your results set.

Object Name:

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 Lock User Go

Select	UserName	Account Status	Created	Default Tablespace	Temporary Tablespace	Profile	Created
<input type="checkbox"/>	ANONYMOUS	EXPIRED & LOCKED	Aug 3, 2007 1:34:38 AM MDT	SYSAUX	TEMP	DEFAULT	Aug 3, 2007 1:34:38 AM MDT
<input type="checkbox"/>	APEX_PUBLIC_USER	EXPIRED & LOCKED	Aug 3, 2007 2:04:08 AM MDT	USERS	TEMP	DEFAULT	Aug 3, 2007 2:04:08 AM MDT
<input type="checkbox"/>	BI	EXPIRED & LOCKED	Aug 4, 2008 7:04:49 PM MDT	USERS	TEMP	DEFAULT	Aug 4, 2008 7:04:49 PM MDT

유저를 선택하고 Unlock User를 선택하고 Go를 누릅니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

유저 계정 Lock 해제 및 암호 재설정

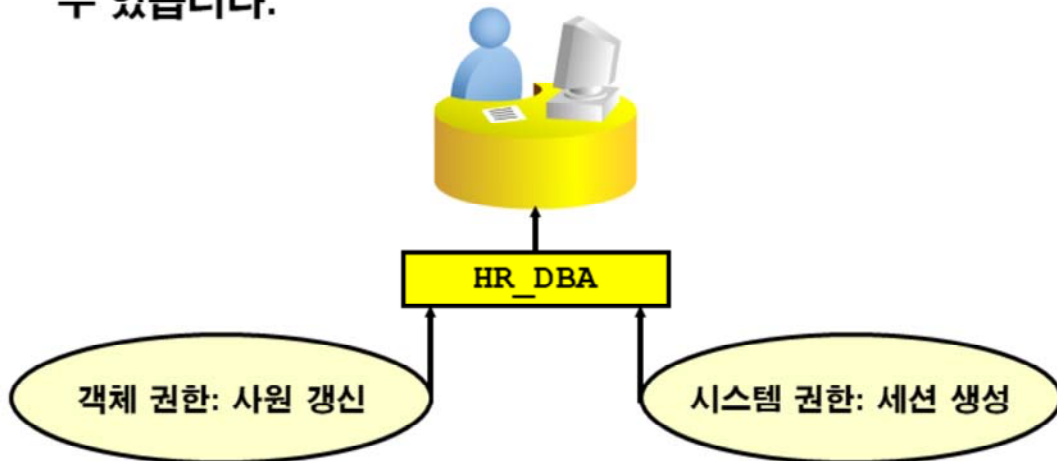
설치 및 데이터베이스 생성 시 Oracle에서 제공하는 많은 데이터베이스 유저 계정의 lock을 해제하고 재설정할 수 있습니다. 이때 유저 계정의 lock을 해제하지 않은 경우 나중에 Users 페이지에서 유저를 선택하고 Actions list에서 Unlock User를 선택한 다음 Go를 눌러서 유저의 Lock을 해제할 수 있습니다. 이렇게 해도 암호는 변경되지 않습니다. 유저의 Lock을 해제했을 때 암호가 만료된 경우 유저를 편집하고 암호를 변경할 때까지 암호가 만료된 상태로 유지됩니다. 유저의 lock을 해제하고 암호를 재설정하려면 Edit Users 페이지에서 다음 단계를 수행합니다.

1. Enter Password 및 Confirm Password 필드에 새 암호를 입력합니다.
2. Unlocked 체크 박스를 선택합니다.
3. Apply를 눌러 암호를 재설정하고 유저 계정의 lock을 해제합니다.

권한

다음과 같은 두 가지 유형의 유저 권한이 있습니다.

- **시스템 권한:** 유저가 데이터베이스에서 특정 작업을 수행할 수 있도록 합니다.
- **객체 권한:** 유저가 특정 객체를 액세스 및 조작할 수 있습니다.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

권한

*권한(privilege)*은 특정 유형의 SQL 문을 실행하거나 다른 유저의 객체에 액세스할 수 있는 권한입니다. 오라클 데이터베이스를 사용하면 데이터베이스 내에서 유저가 수행할 수 있거나 수행할 수 없는 작업을 제어할 수 있습니다.

권한은 두 개의 범주로 나뉩니다.

- **시스템 권한:** 유저는 각 시스템 권한을 사용하여 특정 데이터베이스 작업이나 특정 유형의 데이터베이스 작업을 수행할 수 있습니다. 예를 들어, 테이블스페이스 생성 권한은 시스템 권한입니다. 시스템 권한은 관리자나 권한을 관리할 수 있는 권한을 명시적으로 받은 유저가 부여할 수 있습니다. 170가지 이상의 시스템 권한이 있습니다. 많은 시스템 권한에는 ANY 절이 포함됩니다.
- **객체 권한:** 유저는 객체 권한을 사용하여 테이블, 뷰, 시퀀스, 프로시저, 함수 또는 패키지 등의 특정 객체에 대해 특정 작업을 수행할 수 있습니다. 권한이 없는 유저는 자신이 소유한 객체에만 액세스할 수 있습니다. 객체 권한은 객체의 소유자, 관리자 또는 객체에 대한 권한을 부여할 수 있는 권한을 명시적으로 받은 유저가 부여할 수 있습니다.

시스템 권한

Edit User: HR

Actions: Create Like Go Show SQL Revert Apply

General Roles **System Privileges** Object Privileges Quotas Consumer Group Privileges Proxy Users

System Privilege **Admin Option**

System Privilege	Admin Option
ALTER SESSION	<input type="checkbox"/>
CREATE DATABASE LINK	<input type="checkbox"/>
CREATE SEQUENCE	<input type="checkbox"/>
CREATE SESSION	<input type="checkbox"/>
CREATE SYNONYM	<input type="checkbox"/>
CREATE VIEW	<input type="checkbox"/>
UNLIMITED TABLESPACE	<input type="checkbox"/>

Modify System Privileges Cancel OK

Available System Privileges

- ACCESS_ANY_WORKSPACE
- ADMINISTER ANY SQL TUNING SET
- ADMINISTER DATABASE TRIGGER
- ADMINISTER RESOURCE MANAGER
- ADMINISTER SQL MANAGEMENT OBJECT
- ADMINISTER SQL TUNING SET
- ADVISOR
- ALTER ANY ASSEMBLY
- ALTER ANY CLUSTER
- ALTER ANY CUBE

Move Move All Remove Remove All

Selected System Privileges

- ALTER SESSION
- CREATE DATABASE LINK
- CREATE SEQUENCE
- CREATE SESSION
- CREATE SYNONYM
- CREATE VIEW
- UNLIMITED TABLESPACE

ORACLE

Copyright © 2010, Oracle. All rights reserved.

시스템 권한

시스템 권한을 부여하려면 Edit User 페이지에서 Systems Privileges 탭을 누릅니다. 사용 가능한 권한 리스트에서 알맞은 권한을 선택하고 Move 화살표를 눌러 Selected System Privileges list로 이동합니다.

ANY 절을 사용하면 해당 스키마의 범위를 벗어나 권한을 부여할 수 있습니다. 예를 들어, CREATE TABLE 권한이 있는 경우에는 테이블을 자신의 스키마 내에만 생성할 수 있습니다. SELECT ANY TABLE 권한은 다른 사용자가 소유한 테이블에서 선택할 수 있습니다. SYS 유저와 DBA 롤을 소유한 유저에게는 모든 ANY 권한이 부여되므로 모든 데이터 객체에 대해 원하는 작업을 모두 수행할 수 있습니다. ANY 시스템 권한의 범위는 Oracle Database Vault Option을 사용하여 제어할 수 있습니다.

Admin Option 체크 박스를 선택하면 해당 유저가 권한을 관리하고 시스템 권한을 다른 유저에게 부여할 수 있습니다.

시스템 권한을 부여하기 위한 SQL 구문은 다음과 같습니다.

```
GRANT <system_privilege> TO <grantee clause> [WITH ADMIN OPTION]
```

시스템 권한을 부여하기 전에 보안 요구 사항을 주의 깊게 고려해야 합니다. 일부 시스템 권한은 일반적으로 관리자에게만 부여됩니다.

- **RESTRICTED SESSION:** 이 권한을 사용하면 데이터베이스가 제한 모드로 열려 있어도

로그인할 수 있습니다.

시스템 권한(계속)

- **SYSDBA 및 SYSOPER:** 이 권한을 사용하면 데이터베이스에서 종료, 시작, recovery 및 다른 관리 작업을 수행할 수 있습니다. SYSOPER는 사용자가 기본적인 작업을 수행할 수 있지만 유저 데이터를 볼 수 있는 권한은 없습니다. 다음 시스템 권한을 포함합니다.
 - STARTUP 및 SHUTDOWN
 - CREATE SPFILE
 - ALTER DATABASE OPEN/MOUNT/BACKUP
 - ALTER DATABASE ARCHIVELOG
 - ALTER DATABASE RECOVER (Complete Recovery의 경우만 해당됩니다. UNTIL TIME | CHANGE | CANCEL | CONTROLFILE과 같은 모든 유형의 Incomplete Recovery에는 SYSDBA로 연결해야 함)
 - RESTRICTED SESSION

SYSDBA 시스템 권한에는 Incomplete Recovery 및 데이터베이스 삭제 권한이 추가로 부여됩니다. SYSDBA 시스템 권한을 통해 유저는 SYS 유저로 연결할 수 있습니다.

- **SYSASM:** 이 권한을 사용하면 ASM Instance를 시작, 종료 및 관리할 수 있습니다.
- **DROP ANY object:** DROP ANY 권한을 사용하면 다른 스키마 유저가 소유한 객체를 삭제할 수 있습니다.
- **CREATE, MANAGE, DROP 및 ALTER TABLESPACE:** 이러한 권한을 사용하면 테이블스페이스 속성 생성, 삭제 및 변경과 같은 테이블스페이스 관리를 수행할 수 있습니다.
- **CREATE LIBRARY:** 오라클 데이터베이스에서는 개발자가 PL/SQL 내에서 C 라이브러리와 같은 외부 코드를 생성 및 호출할 수 있습니다. 라이브러리의 이름은 데이터베이스의 LIBRARY 객체가 지정해야 합니다. CREATE LIBRARY 권한을 사용하면 유저가 PL/SQL에서 실행할 수 있는 임의 코드 라이브러리를 생성할 수 있습니다.
- **CREATE ANY DIRECTORY:** 보안을 위해서 코드가 있는 운영 체제 디렉토리는 가상 오라클 디렉토리 객체에 연결되어야 합니다. CREATE ANY DIRECTORY 권한을 사용하면 비보안 코드 객체를 호출할 수 있습니다. CREATE ANY DIRECTORY 권한을 사용하면 Oracle 소프트웨어 소유자가 액세스할 수 있는 임의의 디렉토리에 유저가 읽기 및 쓰기 권한이 있는 디렉토리 객체를 생성할 수 있습니다. 즉, 유저가 해당 디렉토리의 외부 프로시저에 액세스할 수 있습니다. 유저가 데이터 파일, 리두 로그 및 감사(audit) 로그와 같은 데이터베이스 파일에 대해 직접 읽기/쓰기를 시도할 수 있습니다. 조직에 이와 같은 강력한 권한의 남용을 방지하는 보안 전략이 있어야 합니다.
- **GRANT ANY OBJECT PRIVILEGE:** 이 권한은 자신이 소유하지 않은 객체에 대해 객체 사용 권한을 부여할 수 있습니다.
- **ALTER DATABASE 및 ALTER SYSTEM:** 매우 강력한 권한으로, 데이터 파일 이름 변경 또는 버퍼 캐시 비우기와 같은 데이터베이스 및 Oracle Instance 수정 작업을 수행할 수 있습니다.

객체 권한

객체 권한을 부여하려면 다음을 수행하십시오.

- 객체 유형을 선택합니다.
- 객체를 선택합니다.
- 권한을 선택합니다.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

객체 권한

객체 권한을 부여하려면 Edit User 페이지에서 Object Privileges 탭을 누릅니다. 권한을 부여할 객체 유형을 선택하고 Add 버튼을 누릅니다. <username.object name>을 입력하거나 리스트에서 선택하는 방식으로 객체를 선택합니다.

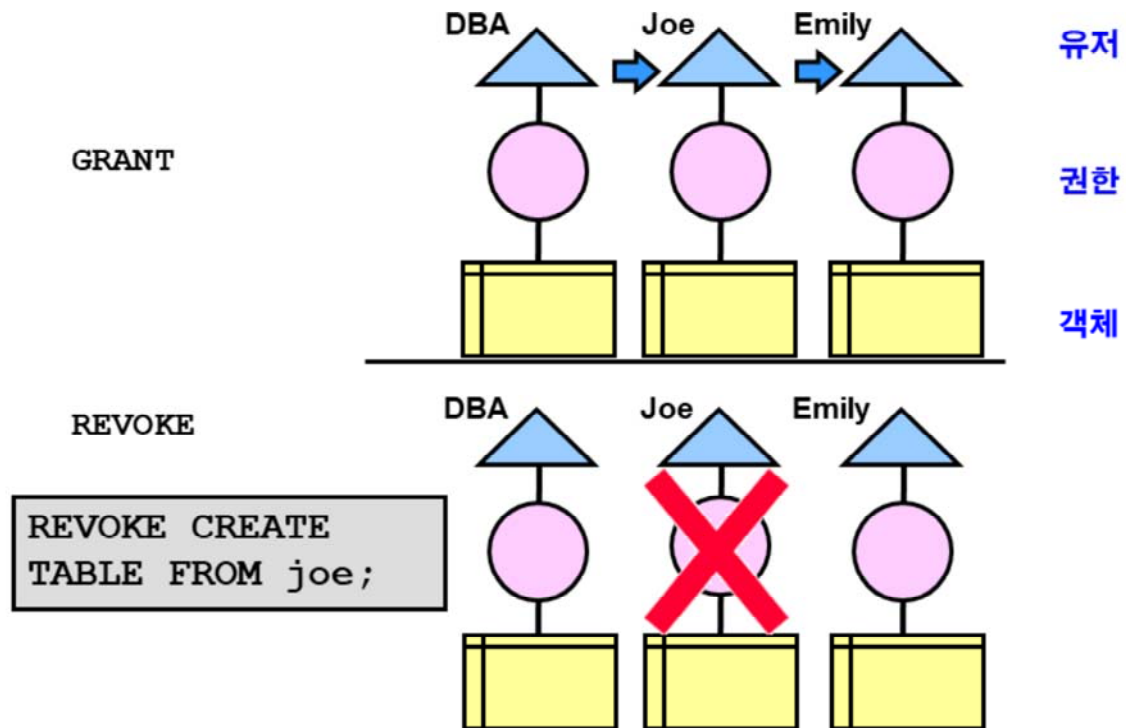
그런 다음 Available Privileges list에서 적합한 권한을 선택하고 Move 버튼을 누릅니다. 권한 선택을 마친 후에 OK를 누릅니다.

Edit User 페이지에서 이 유저가 다른 유저에게 동일한 액세스 권한을 부여할 수 있는 경우 Grant 체크 박스를 선택합니다.

객체 권한을 부여하기 위한 SQL 구문은 다음과 같습니다.

```
GRANT <object_privilege> ON <object> TO <grantee clause>
[WITH GRANT OPTION]
```


ADMIN OPTION을 사용하여 시스템 권한 취소



ORACLE

Copyright © 2010, Oracle. All rights reserved.

ADMIN OPTION을 사용하여 시스템 권한 취소

GRANT 명령으로 직접 부여된 시스템 권한은 REVOKE SQL 문을 사용하여 취소할 수 있습니다. 시스템 권한에 대해 ADMIN OPTION을 가진 유저는 다른 데이터베이스 유저의 권한을 취소할 수 있습니다. 이 작업은 원래 권한을 부여한 유저가 아니어도 가능합니다.

ADMIN OPTION이 부여되었는지 여부에 관계없이 시스템 권한을 취소할 때는 연쇄적으로 취소되지 않습니다.

시스템 권한을 취소하기 위한 SQL 구문은 다음과 같습니다.

```
REVOKE <system_privilege> FROM <grantee clause>
```

이 슬라이드에서는 다음과 같은 상황을 보여줍니다.

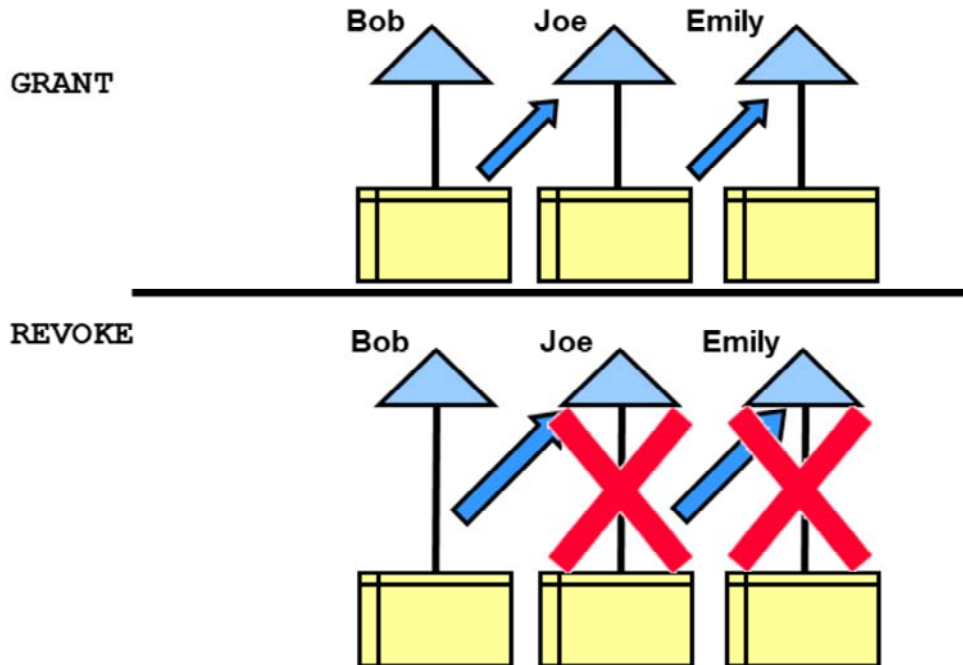
시나리오

1. DBA가 ADMIN OPTION을 사용하여 CREATE TABLE 시스템 권한을 Joe에게 부여합니다.
2. Joe가 테이블을 생성합니다.
3. Joe가 CREATE TABLE 시스템 권한을 Emily에게 부여합니다.
4. Emily가 테이블을 생성합니다.
5. DBA가 Joe에게 부여된 CREATE TABLE 시스템 권한을 취소합니다.

결과

Joe의 테이블은 여전히 존재하지만 Joe는 새 테이블을 생성할 수 없습니다. Emily의 테이블은 여전히 존재하며 Emily는 CREATE TABLE 시스템 권한을 갖습니다.

GRANT OPTION을 사용하여 객체 권한 취소



ORACLE

Copyright © 2010, Oracle. All rights reserved.

GRANT OPTION을 사용하여 객체 권한 취소

DML(데이터 조작어) 작업과 관련된 시스템 권한을 취소할 때 연쇄적인 결과를 관찰할 수 있습니다. 예를 들어, 유저에게 `SELECT ANY TABLE` 권한이 부여되고 해당 유저가 테이블을 사용하는 프로시저를 생성한 경우 유저의 스키마에 포함된 모든 프로시저를 다시 사용하려면 재컴파일해야 합니다.

객체 권한 취소는 `GRANT OPTION`이 제공되는 경우에도 연쇄적인 결과를 나타냅니다. 유저는 자신이 부여한 권한만 취소할 수 있습니다. 예를 들어, Bob은 Joe가 Emily에게 부여한 객체 권한을 취소할 수 없습니다. 권한을 부여 받은 사람 또는 `GRANT ANY OBJECT PRIVILEGE`라고 부르는 권한이 있는 유저만 객체 권한을 취소할 수 있습니다.

시나리오

1. `GRANT OPTION`을 사용하여 `EMPLOYEES`에 대한 `SELECT` 객체 권한이 Joe에게 부여됩니다.
2. Joe가 `EMPLOYEES`에 대한 `SELECT` 권한을 Emily에게 부여합니다.
3. Joe에게서 `SELECT` 권한이 취소됩니다. 이 취소는 Emily에게도 연쇄적인 영향을 미칩니다.

롤 사용 시의 이점

- 권한 관리 용이성
- 동적 권한 관리
- 권한의 선택적 가용성



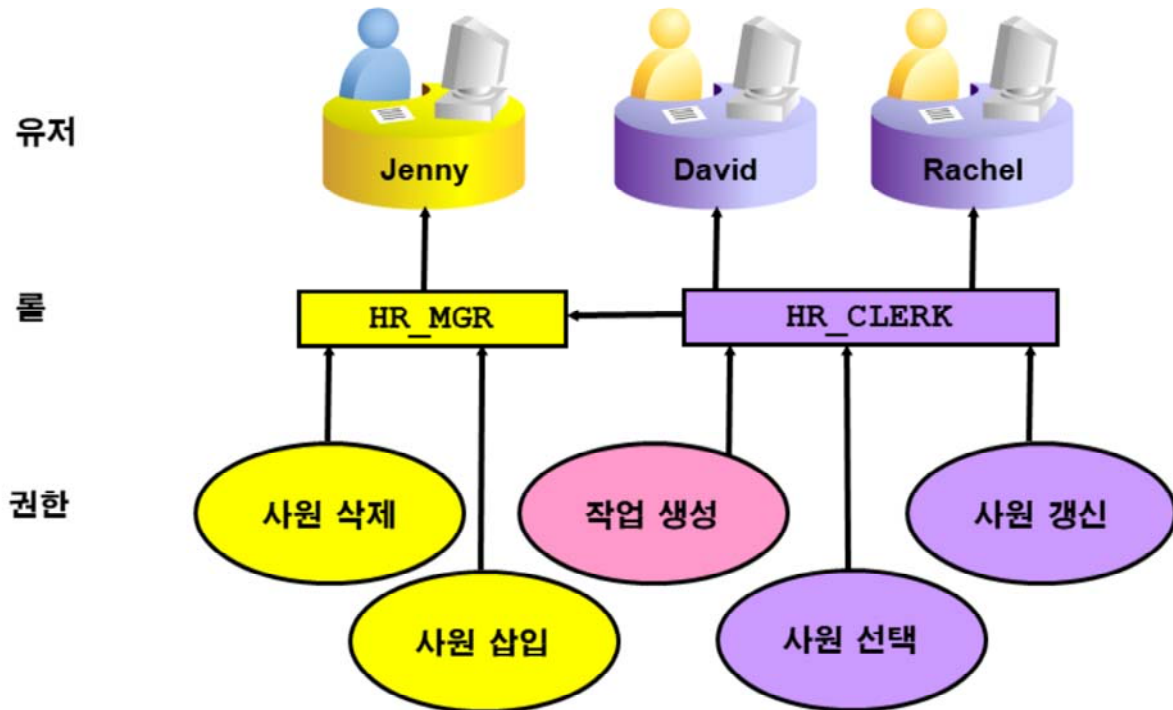
ORACLE

Copyright © 2010, Oracle. All rights reserved.

롤 사용 시의 이점

- **권한 관리 용이성:** 롤을 사용하여 권한 관리를 단순화합니다. 여러 유저에게 일련의 동일한 권한을 부여하는 대신 권한을 롤에 부여한 다음 이 롤을 각 유저에게 부여할 수 있습니다.
- **동적 권한 관리:** 롤과 연관된 권한을 수정하면 그 롤을 부여 받은 모든 유저는 자동으로 즉시 수정된 권한을 얻게 됩니다.
- **권한의 선택적 가용성:** 롤을 활성화 및 비활성화하여 권한을 일시적으로 설정 및 해제할 수 있습니다. 그러면 특정 상황에서 유저의 권한을 제어할 수 있습니다.

롤에 권한 할당 및 유저에게 롤 할당



ORACLE

Copyright © 2010, Oracle. All rights reserved.

롤에 권한 할당 및 유저에게 롤 할당

대부분의 시스템에서는 각 유저에 대해 개별적으로 필요한 권한을 부여하려면 시간이 많이 걸리고 오류가 발생하기 쉽습니다. Oracle 소프트웨어는 롤을 통해 권한 관리를 쉽게 제어할 수 있습니다. 롤은 유저나 다른 롤에 부여된 관련 권한의 명명된 그룹이며 데이터베이스의 권한 관리를 용이하게 하여 보안을 향상시키도록 고안되었습니다.

롤 특성

- 유저와 마찬가지로 롤에도 권한을 부여하고 취소할 수 있습니다.
- 시스템 권한과 마찬가지로 롤을 유저나 다른 롤에 부여하거나 취소할 수 있습니다.
- 시스템 및 객체 권한으로 구성될 수 있습니다.
- 롤을 부여 받은 각 유저에 대해 활성화 및 비활성화될 수 있습니다.
- 롤을 활성화하려면 암호가 필요할 수 있습니다.
- 롤은 유저가 소유하지 않으며 스키마에 존재하지 않습니다.

슬라이드 예제에서 employees 테이블에 대한 SELECT 및 UPDATE 권한과 CREATE JOB 시스템 권한은 HR_CLERK 롤에 부여됩니다. employees 테이블에 대한 DELETE 및 INSERT 권한과 HR_CLERK 롤은 HR_MGR 롤에 부여됩니다. 관리자는 HR_MGR 롤을 부여 받고 이제 employees 테이블을 선택, 삭제, 삽입 및 갱신할 수 있습니다.

미리 정의된 롤

롤	포함된 권한
CONNECT	CREATE SESSION
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
DBA	대부분의 시스템 권한 및 기타 여러 가지 롤. 비관리자에게는 부여 안함.
SELECT_ CATALOG_ ROLE	시스템 권한은 없지만 HS_ ADMIN_ ROLE 및 데이터 디렉터리와 대한 1,700개 이상의 객체 권한이 있음

ORACLE

Copyright © 2010, Oracle. All rights reserved.

미리 정의된 롤

데이터베이스 생성 스크립트를 실행할 때 오라클 데이터베이스에 대해 여러 가지 롤이 자동으로 정의됩니다. CONNECT는 Enterprise Manager를 사용하여 생성된 유저에게 자동으로 부여됩니다. 보안상의 이유로 오라클 데이터베이스 10.2.0 이후 버전에는 CONNECT 롤에 CREATE SESSION 권한만이 포함됩니다.

주: RESOURCE 롤 부여 시 UNLIMITED TABLESPACE 권한 부여가 포함됩니다.

기능 롤

특수 기능 관리 권한을 부여하는 기타 롤은 해당 기능 설치 시 생성됩니다. 예를 들어, XDBADMIN에는 해당 기능이 설치된 경우 XML(Extensible Markup Language) 데이터베이스 관리에 필요한 권한이 포함되어 있습니다. AQ_ADMINISTRATOR_ROLE은 Advanced Queuing에 필요한 권한을 제공합니다. HS_ADMIN_ROLE에는 이기종 서비스 관리에 필요한 권한이 포함되어 있습니다.

실수로 필요한 기능을 비활성화할 수 있기 때문에 이러한 기능 롤에 부여된 권한을 Oracle 지원 센터의 도움 없이 변경해서는 안됩니다.

롤 생성

Server > Roles를 선택합니다.

적합한 탭에서 권한 및 롤을 추가합니다.

완료되면 OK를 누릅니다.

There is no authentication.

Select	Object Privilege	Schema	Object
<input type="radio"/>	SELECT	OE	CUSTOMERS
<input type="radio"/>	SELECT	OE	INVENTORIES
<input type="radio"/>	SELECT	OE	ORDERS
<input type="radio"/>	SELECT	OE	ORDER_ITEMS

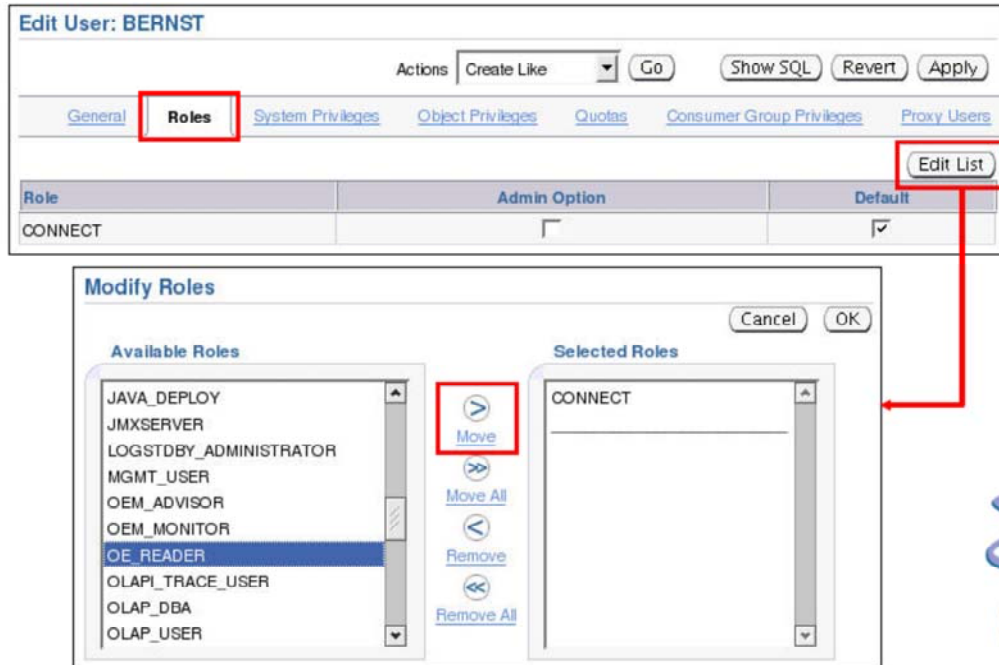
롤 생성

롤(role)은 유저나 다른 롤에 부여된 관련 권한의 명명된 그룹입니다. DBA는 롤을 통해 권한을 관리합니다.

롤을 생성하려면 다음 단계를 수행하십시오.

1. Enterprise Manager Database Control에서 Server 탭을 누르고 Security 머리글 아래에서 Roles를 누릅니다.
2. Create 버튼을 누릅니다.
3. General 탭에서 롤 이름을 입력합니다.
4. 선택적으로 시스템 권한, 객체 권한 및 기타 롤을 필요에 따라 추가합니다. 필요한 경우 나중에 롤을 편집하여 이러한 설정을 수정할 수 있습니다.
5. 완료되면 OK를 누릅니다.

유저에게 롤 할당



ORACLE

Copyright © 2010, Oracle. All rights reserved.

유저에게 롤 할당

롤을 사용하여 데이터베이스 권한을 관리할 수 있습니다. 롤에 권한을 추가한 다음 이 롤을 유저에게 부여할 수 있습니다. 그러면 유저가 롤을 활성화하여 롤이 부여한 권한을 행사할 수 있습니다. 롤에는 이 롤에 부여된 모든 권한과 이 롤에 부여된 다른 롤의 모든 권한이 포함되어 있습니다.

Enterprise Manager에서는 기본적으로 새 유저에게 CONNECT 롤을 자동으로 부여합니다. 이를 통해 유저가 데이터베이스에 연결할 수 있고 유저가 소유한 스키마에 데이터베이스 객체를 생성할 수 있습니다.

유저에게 롤을 할당하려면 다음을 수행하십시오.

1. Enterprise Manager Database Control에서 Server 탭을 누르고 Security 머리글 아래에서 Users를 누릅니다.
2. 유저를 선택하고 Edit 버튼을 누릅니다.
3. Roles 탭을 누른 다음 Edit List 버튼을 누릅니다.
4. Available Roles에서 원하는 롤을 선택하고 Selected Roles로 이동합니다.
5. 적합한 롤을 모두 할당한 후에 OK 버튼을 누릅니다.

퀴즈

Oracle Database 11g 에서 생성한 모든 암호는 기본적으로 대소문자를 구분하지 않습니다.

1. 참
2. 거짓

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 2

퀴즈

데이터베이스 롤의 특성은 무엇입니까?

1. 활성화하거나 비활성화할 수 있음
2. 시스템 및 객체 권한으로 구성할 수 있음
3. 생성자에 의해 소유됨
4. 암호로 보호할 수 없음

ORACLE

Copyright © 2010, Oracle. All rights reserved.

해답: 1, 2

요약

이 단원에서는 다음 항목에 대해 설명했습니다.

- 데이터베이스 유저 계정 생성 및 관리:
 - 유저 인증
 - 기본 저장 영역(테이블스페이스) 할당
- 권한 부여 및 취소
- 롤 생성 및 관리

ORACLE

Copyright © 2010, Oracle. All rights reserved.