

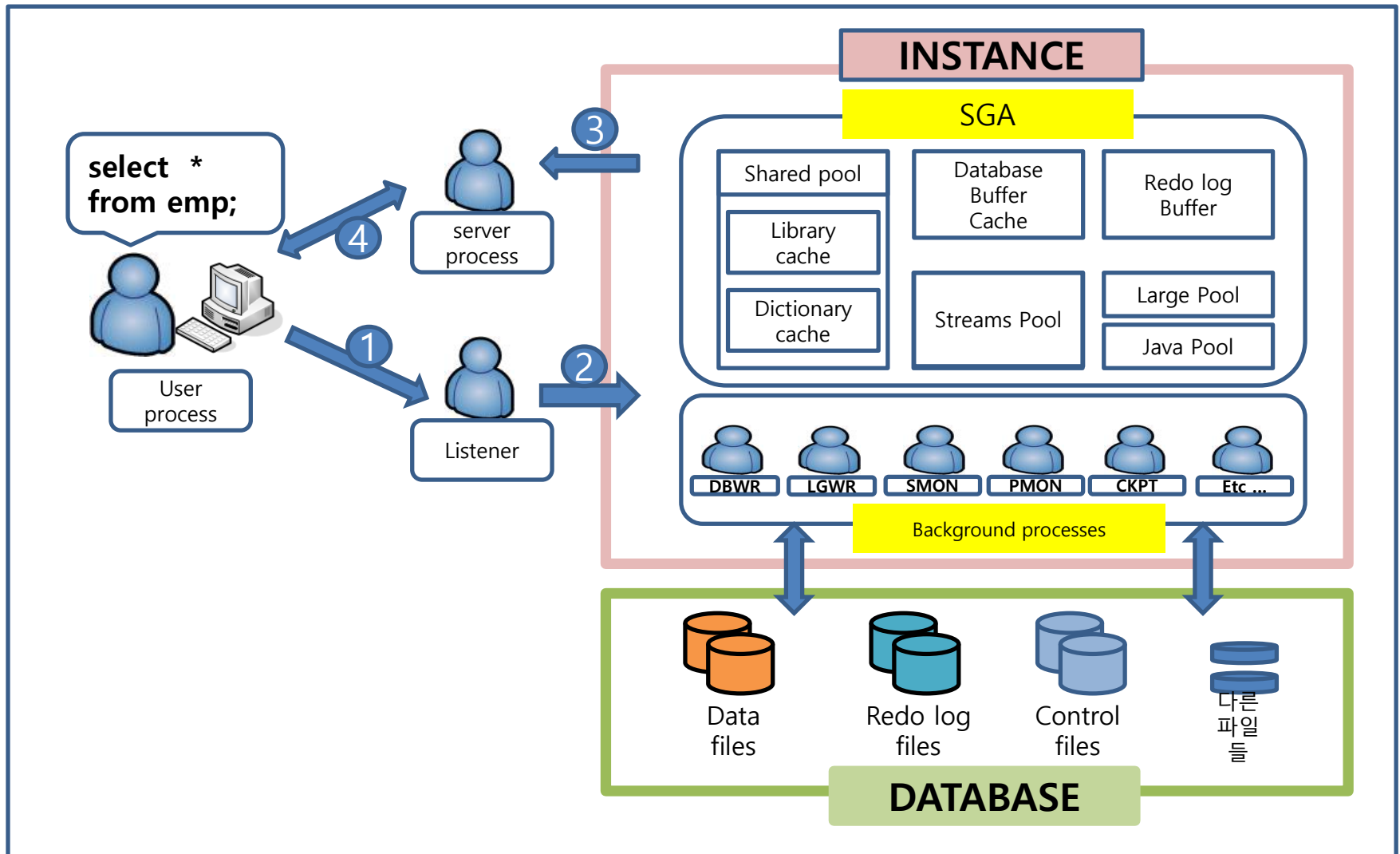


# Part 1 . **Oracle Architecture**

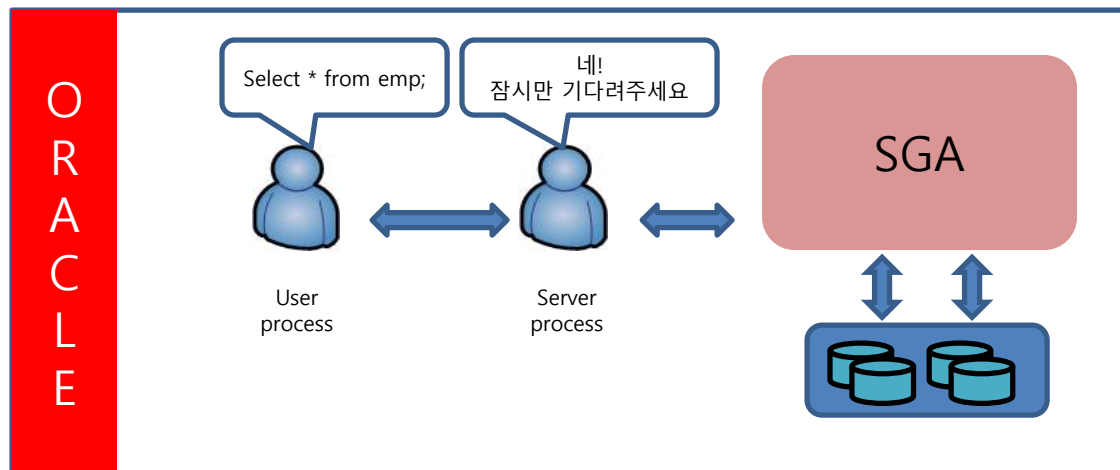
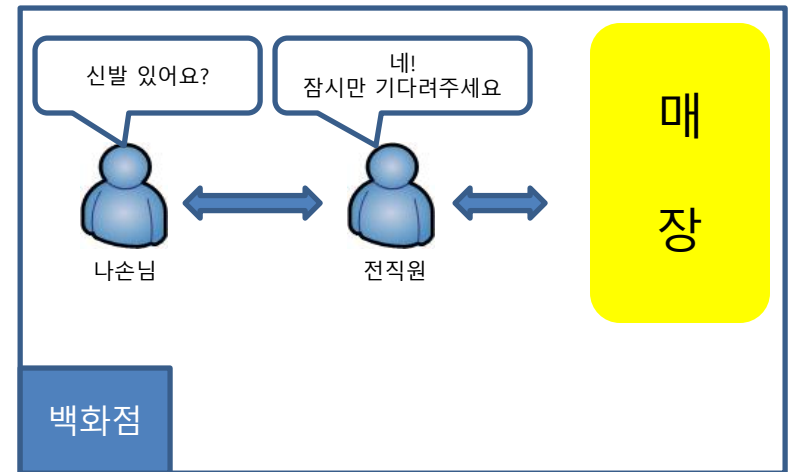
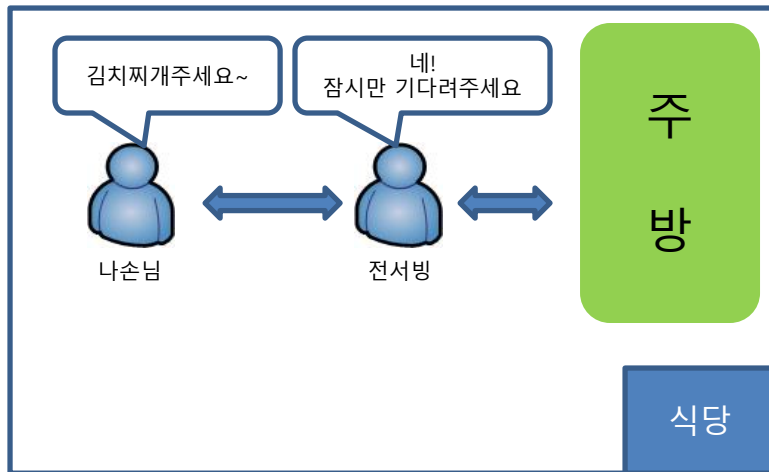
# 1. Oracle Architecture

- oracle server = 인스턴스 + 데이터베이스
- 인스턴스 : 메모리 상에 생성되는 구조
  - ✓ SGA(System Global Area)
  - ✓ 백그라운드 프로세스
- 데이터베이스 : 디스크 영역에 저장되는 구조
  - ✓ Data files : 데이터가 저장되는 영역
  - ✓ Control files : DB 전체의 관리정보가 들어있는 영역
  - ✓ Redo log files : 장애 복구 시에 사용되는 영역

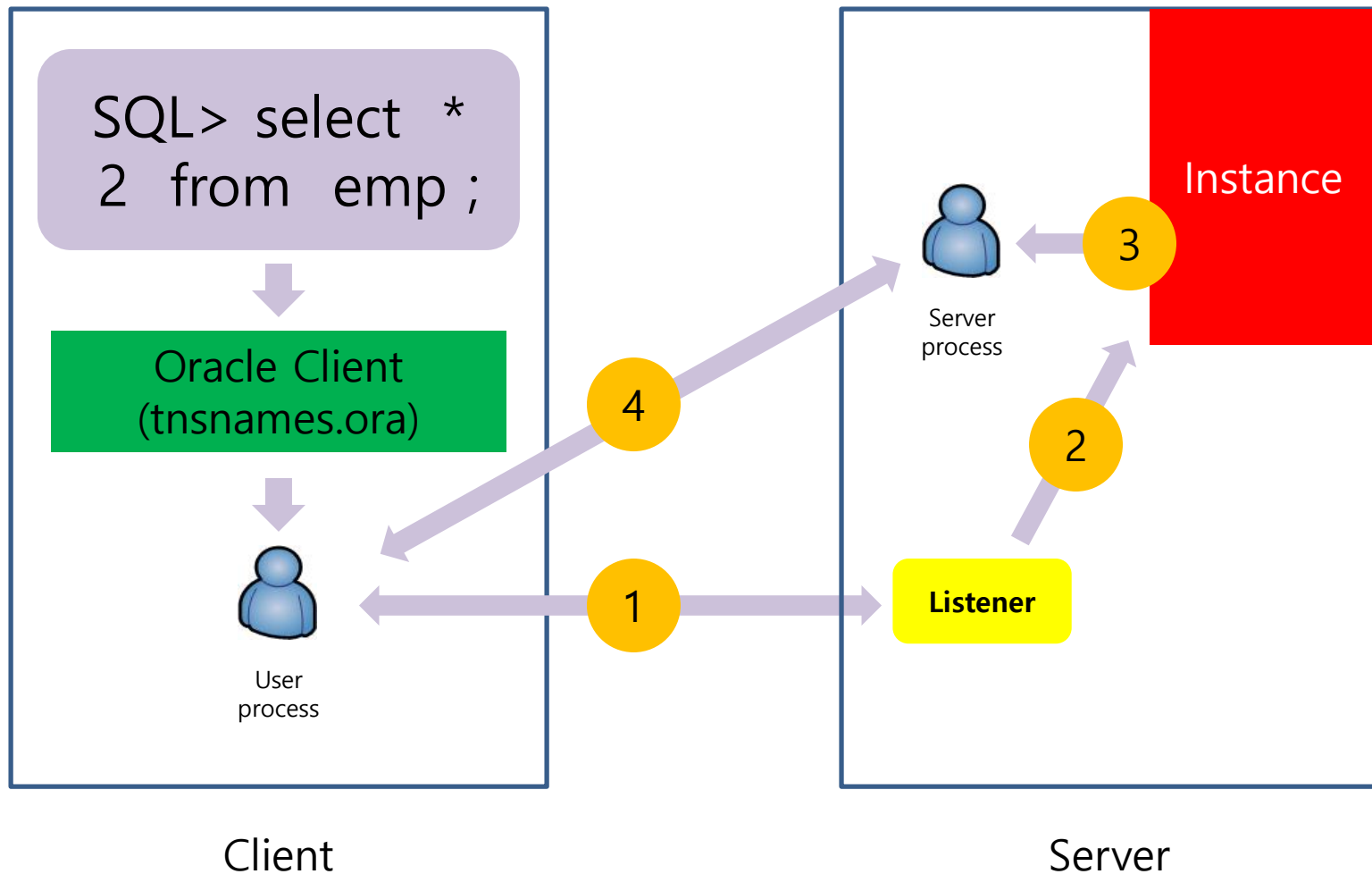
# 1. Oracle Architecture



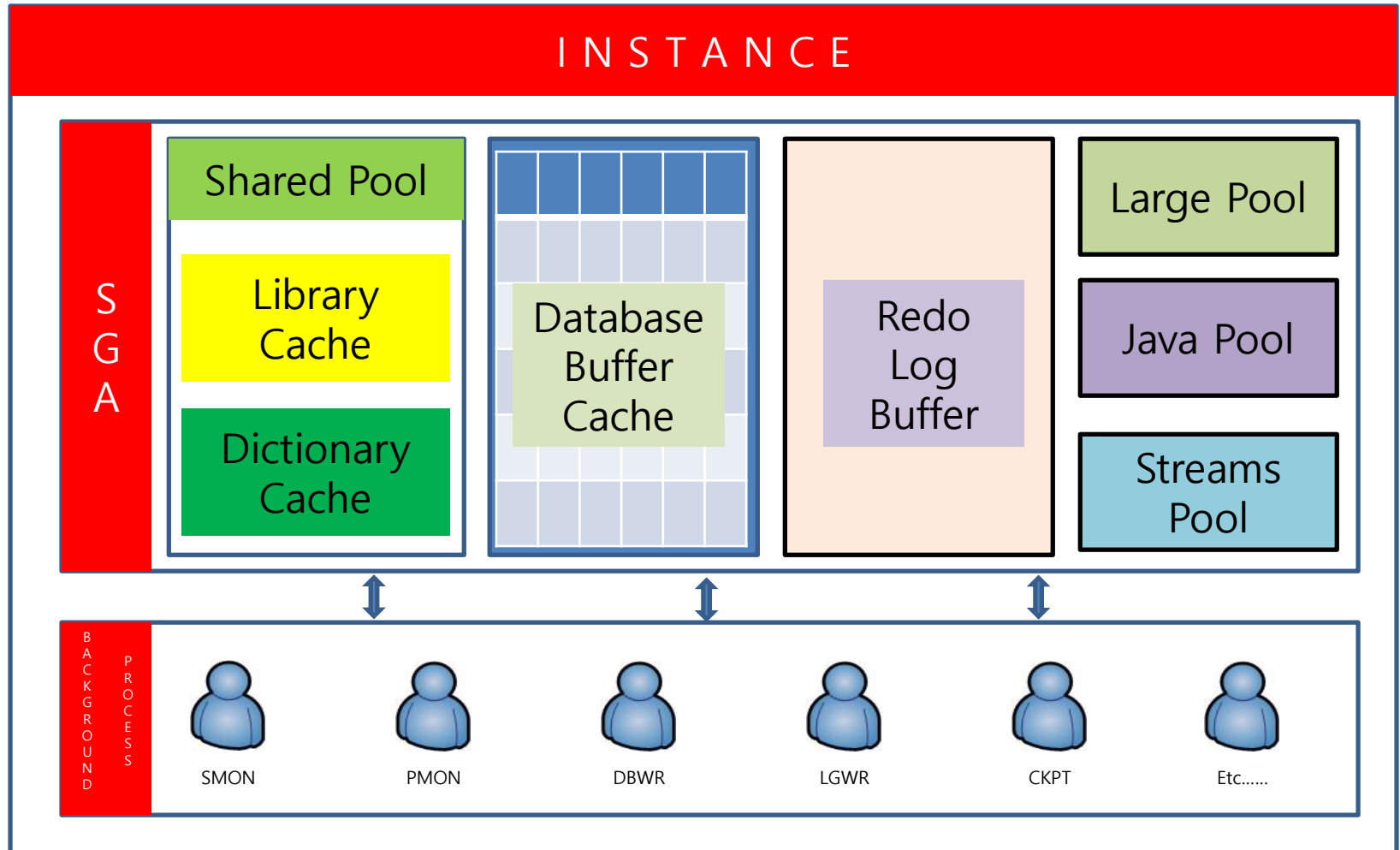
# 1. Oracle Architecture



# 1. Oracle Architecture



# 1. Oracle Architecture



# 1. Oracle Architecture

- SGA란
  - 오라클서버의 메모리영역
  - Oracle 인스턴스에 대한 데이터와 제어 정보를 가지는 공유 메모리 영역의 집합
  - 오라클 파라미터(pfile, spfile) 파일의 조정으로 SGA의 각 부분의 크기를 조절
  - Oracle9i부터 동적으로 변경 가능 (SGA\_MAX\_SIZE내)
  - Shared Pool, DataBase Buffer Cache, Redo Log Buffer와 Large Pool과 Java Pool로 구성

# 1. Oracle Architecture

## 1. Shared Pool

- Shared Pool은 Library Cache와 Data Dictionary Cache로 구성
- Shared Pool은 하나의 데이터베이스에 행해지는 모든 SQL 문을 처리하기 위하여 사용
- Shared Pool은 문장을 실행하기 위해 그 문장과 관련된 실행 계획과 구문분석 정보가 들어있음
- SHARED\_POOL\_SIZE 파라미터 값으로 결정



# 1. Oracle Architecture

## 1) Library Cache

- 가장 최근에 사용된 SQL 문장의 명령문과, 구문 분석 트리, 실행계획 정보를 가지고 있습니다.
- LRU(Least Recently Used) 알고리즘으로 관리
- 동일한 문장이 다음 번에 실행되면 Shared SQL 영역에 저장되어 있는 실행계획과 파싱 트리를 그대로 이용하기 때문에 SQL 문장의 처리 속도가 향상(Soft Parsing)

# 1. Oracle Architecture

## 2) Data Dictionary Cache

- 테이블, 컬럼, 사용자 이름, 사용 권한 같은 가장 최근에 사용된 데이터 사전의 정보를 저장
- 구문 분석 단계에서 서버 프로세스는 SQL문에 지정된 오브젝트 이름을 찾아내고 접근 권한을 검증하기 위해 Dictionary Cache의 정보를 찾아봄

# 1. Oracle Architecture

## 3) Server Result Cache(11g New Feature)

- SQL 실행의 결과 값을 cache해 두는 공간
- SQL 수행 시 해당 데이터가 DB Buffer Cache에 있는지 찾는 과정에서 latch가 발생되는데 이를 해결하기 위해 그 결과값을 Shared Pool영역의 Server Result Cache에 저장
- 동일한 Select가 수행되었을 경우 DB Buffer Cache까지 가지 않고 즉시 Server Result Cache에서 가져갈 수 있도록 함

# 1. Oracle Architecture

## 3) Shared Pool Size

- 확인

```
SQL> show parameter shared_pool_size; <프롬프트창>
```

```
SQL> select *  
      from v$parameter  
      where name ='shared_pool_size'; <접속툴:Orange>
```

- 변경

```
SQL> alter system set shared_pool_size=10m;
```

# 1. Oracle Architecture

## 2. Database Buffer Cache(Buffer Cache)

- 디스크에서 읽은 데이터 블록의 복사본을 가지고 있다. 동시 접속된 사용자들은 Database Buffer Cache를 공유 하여 Access 한다.
- LRU 알고리즘에 의하여 가장 오래 전에 사용된 것은 디스크에 저장 하고 메모리에는 가장 최근에 사용된 데이터를 저장 함으로, 디스크 입출력이 줄어 들고, 데이터베이스 시스템의 성능이 증가 됨
- 데이터를 조회 -> 해당 데이터를 먼저 Database Buffer Cache에서 찾고 있으면 반환(Logical Read), 없으면 Database Buffer Cache의 Free Buffer를 일단 확보 후 Disk에서 데이터를 읽어 들여 cache하여 반환(Physical Read)

# 1. Oracle Architecture

## 2. Database Buffer Cache(Buffer Cache)

- Database Buffer Cache Block의 상태
  - Pinned Buffer : 다른 사용자가 이미 사용하고 있는 Buffer Block으로 사용할 수 없음
  - Dirty Buffer : 현재 작업은 진행되지 않지만 다른 사용자가 내용을 변경한 후 아직 데이터 파일에 변경된 내용을 저장 하지 않은 Buffer
  - Free Buffer : 이 Buffer는 사용되지 않았거나(Unused) 또는 Dirty Buffer 였다가 디스크로 저장이 되고 다시 재사용 가능하게 된 Block
- LRU List : Buffer Block들의 상태를 관리하고 있는 list
  - 수많은 사용자가 동시에 Physical Read를 하여 동시에 Database Buffer Cache의 Free Buffer를 찾으려고 할 때 이 LRU List를 참조 -> 동시성 관리를 위해 순번 제공(Latch) -> 본인 순번이 올 때까지 대기

# 1. Oracle Architecture

## 2. Database Buffer Cache(Buffer Cache) SIZE

- 확인

SQL) show parameter db\_cache\_size;

SQL) select \*

from v\$parameter

where name ='db\_cache\_size';

- 변경

SQL> alter system set db\_cache\_size=10m;

# 1. Oracle Architecture

## 3. Redo Log Buffer

- 리두 로그 버퍼는 데이터베이스에서 일어난 모든 변화를 저장하는 메모리 공간
- 장애 발생 시 복구를 위해 모든 변경사항 저장
- Redo Log Buffer에 기록하지 않는 경우도 있음
  - Direct Load(SQL Loader, insert /\*+ append \*/)
  - table이나 index의 nologging 옵션의 경우
  - 단, table nologging 시 DML 작업의 경우 제한적으로 Redo log에 기록
- DB에서 발생한 모든 변화는 LGWR에 의해 리두 로그 파일에 저장
- LOG\_BUFFER로 크기 지정



# 1. Oracle Architecture

## 4. Large Pool

- SGA 내부의 대규모 메모리 할당을 제공하기 위한 선택적 공간
- I/O 서버 프로세스 및 병렬 버퍼 할당 시 사용 : backup과 recovery를 위해 RMAN이라는 유틸리티를 사용하는데, 이때 RMAN을 사용하게 되면 여러 개의 디스크 I/O 프로세스를 기동하게 된다. BACKUP\_DISK\_IO=N과 BACKUP\_TAPE\_IO\_SLAVE=TRUE로 설정되어 있을 경우 사용됨
- Parallel 작업 시 사용 : 병렬 작업 수행 시 PARALLEL\_AUTOMATIC\_TUNING=TRUE로 설정시 사용
- LARGE\_POOL\_SIZE 파라미터로 관리되며, 기본 크기는 0

# 1. Oracle Architecture

## 5. Java Pool

- 자바로 작성된 프로그램을 실행할 때 실행 계획을 저장하는 영역
- JAVA\_POOL\_SIZE 파라미터로 관리되며, 기본 크기 24MB가 할당

# 1. Oracle Architecture

## 6. Streams Pool

- 10g New Feature
- Stream 기능 사용할 경우에만 사용되며 기본값은 0임
  - Stream 기능 : 오라클에서 제공하는 DB 동기화 방식

## 7. Fixed SGA

- Oracle이 내부적으로 사용하기 위해 생성시키는 공간
- 주로 백그라운드 프로세스들이 필요한 database 전반적인 공유 정보나 각 프로세스들끼리 공유해야 하는 lock 정보 같은 내용들이 저장
- Oracle이 시작될 때 자동으로 설정되며 사용자나 관리자가 임의로 변경할 수 없음

# 1. Oracle Architecture

- **Dynamic SGA 기능**

- Oracle 9i 이후부터 등장하게 된 SGA 동적 관리 기법
- 파라미터의 크기를 Oracle instance 재기동 없이 변경할 수 있음
- Dynamic SGA를 사용할 수 있는 SGA 관련 파라미터는 DB\_CACHE\_SIZE, SHARED\_POOL\_SIZE, LARGE\_POOL\_SIZE, JAVA\_POOL\_SIZE
- 각 SGA 파라미터의 합은 SGA\_MAX\_SIZE보다 작아야 함
- SGA\_MAX\_SIZE는 동적으로 변경 불가능
- 파라미터 사이즈 할당 시 그라눌 단위로 할당됨

# 1. Oracle Architecture

- **Granule**

- Granule은 가상 메모리 상의 연속된 공간으로, dynamic SGA 모델에서 할당할 수 있는 최소 단위
- 이 Granule의 크기는 SGA\_MAX\_SIZE에 따라 다음과 같이 구분
  - if estimated SGA size is < 1G 4MB else 16MB
- Dynamic SGA를 사용할 수 있는 SGA관련 파라미터는 이 Granule 단위로 늘어나거나 줄어듬

# 1. Oracle Architecture

- **SGA 사이즈 확인**

- Total System Global Area : 전체 SGA 영역
- Fixed Size : 백그라운드 프로세스들이 사용하는 예약된 공간
- Variable Size : Shared Pool, Large Pool, Java Pool 공간
- Database Buffers : Database Buffer Cache 영역
- Redo Buffers : Redo log Buffer 영역

SQL> show sga

Total System Global Area	1670221824 bytes
Fixed Size	2176328 bytes
Variable Size	1056967352 bytes
Database Buffers	603979776 bytes
Redo Buffers	7098368 bytes

# 1. Oracle Architecture

- **PGA(Program Global Area)**

- 하나의 단일 프로세스에 대한 데이터와 제어 정보를 가지고 있는 메모리 공간으로 USER PROCESS가 Oracle Database에 접속하고 Session이 생성될 때 Oracle에 의해 할당(1:1)
- 각 SERVER PROCESS에 하나만 할당되는 PGA 메모리 영역은 SGA 영역과 달리 다른 프로세스와 공유되지 않는, 각 프로세스가 독립적으로 사용하는 non-shared 메모리 영역
- PGA는 프로세스가 생성될 때 할당되며 프로세스가 종료될 때 해제 됩니다.
- PGA\_AGGREGATE\_TARGET라는 값을 통해 사이즈 설정

# 1. Oracle Architecture

## 1) Private SQL Area

- User Process로 부터 전달받은 SQL의 Bind 변수 값 보관
- Query의 실행 상태 정보 및 Query의 임시 정보 저장

## 2) SQL Work Area

- Sort나 Hash관련 작업을 수행하는 공간
- order by나 union과 같은 정렬작업을 필요로 하는 SQL 구문 실행 시 1차적으로 해당 공간에서 작업 수행
- INDEX 생성 작업 역시 정렬이 필요하므로 해당 영역에서 1차적 작업 수행