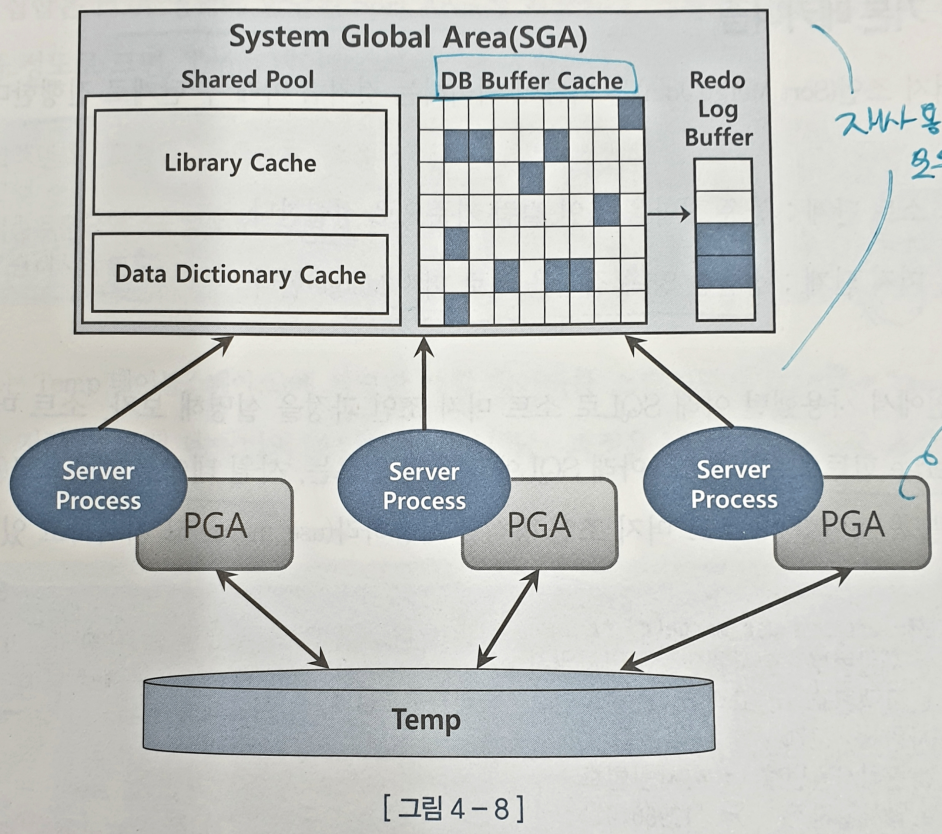
소트 머지 조인

**#** SGA ( System Global Area ) vs PGA ( Private Global Area )



SGA

* 서버 프로세스와 백그라운 프로세스가 공통으로 액세스하는 데이터와 제어 구조를 캐싱하는 메모리 공간.
* **여러 프로세스 공유 가능**, 동시 액세스 불가
* 액세스 직렬화를 위한 Lock 매커니즘 존재 = Latch(래치)

PGA

* **자신만의 고유 메모리 영역**
* 프로세스에 종속적인 고유 데이터를 저장하는 용도
* 공간이 작아 데이터를 저장하지 못하는 경우 Temp 테이블스페이스 사용
* 독립적인 공간으로 같은 양의 데이터일 경우 SGA보다 빠르게 읽을 수 있다.

**#** 소트 머지 조인의 기본 매커니즘

1. 소트단계 : 양쪽 집합을 조인 컬럼 기준으로 정렬
2. 머지 단계 : 정렬한 양쪽 집합을 서로 머지

select /\*+ ordered **use\_merge(c)** \*/

e.사원번호, e.사원명, e.입사일자, c.고객번호, c.고객명, c.전화번호, c.최종주문금액

from 사원 e, 고객 c

where c.관리사원번호 = e.사원번호

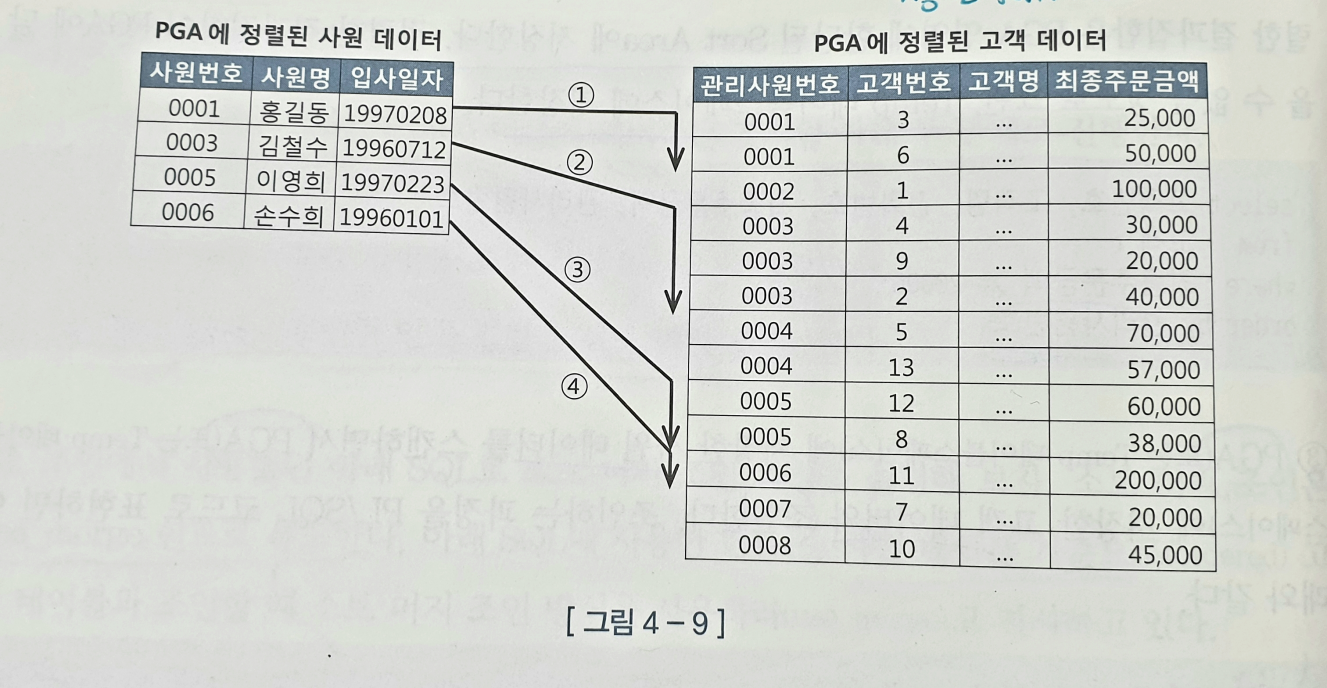
and e.입사일자 >= ‘19960101’

and e.부서코드 = ‘z123’

and c.최종주문금액 >= 20000

( 힌트 – use\_merge 소트 머지 조인 유도 힌트 / ordered 사원테이블 기준 )

SQL 수행과정

1. 조건에 해당하는 사원데이터를 사원번호 순으로 **정렬해서 PGA 영역에 저장**
2. 조건에 해당하는 고객 데이터를 조인컬럼인 관리사원번호 순으로 **정렬해 PGA 영역에 저장**
3. PGA에 저장된 사원 데이터를 스캔하면서 PGA에 저장된 고객 데이터와 **조인** -> **NL 조인과 같음**

* 고객 데이터를 매번 **Full scan 하지 않는다.**
* 고객 데이터가 정렬되어 있어 원하는 지점을 쉽게 찾을 수 있다.
* **데이터 자체가 인덱스 역할을 한다. = 인덱스가 없어도 사용 가능한 조인 방식**
* 대량 데이터 조인시 NL 조인보다 유리하다.

**#** 소트 머지 조인이 빠른 이유

* Sort Area의 미리 정렬된 자료구조 사용
* **PGA 공간이 프로세스만을 위한 독립적인 메모리 공간이므로 래치 획득 과정이 없다**.   
  = 대량 데이터 조인에 유리한 이유
* NL 조인은 모든 인덱스와 테이블이 블록에 래치 획득 및 캐시버퍼에 체인 스캔 과정을 거친다. 버퍼캐시에 없는 블록은 건건이 디스크에서 읽는다.

**#** 소트 머지 조인의 주용도

* 조인 조건식이 등치 (=) 조건이 아닌 대량 데이터 조인  
  ( 등치 조건일 경우 해시 조인이 더 빠르다. )
* 조인 조건식이 아예 없는 조인

**#** 소트 머지 조인 제어하기

* 힌트를 사용해서 제어 ( 280p 참고 )

**#** 소트 머지 조인 특징

* 조인을 위한 실시간 인덱스 생성과 같음
* **소트 부하 감수시, 건건이 버퍼캐시를 경유하는 NL 조인보다 빠름**
* **인덱스 유무에 영향을 받지 않음**
* 양쪽 집합을 개별적으로 읽고 조인 수행   
  = 조인 컬럼에 인덱스가 없는 경우 **두 테이블에서 조인 대상 집합을 줄일 수 있을 때 유리함**
* 스캔 위주의 액세스 방식 사용
* 조인 대상 레코드를 찾을 때 인덱스를 사용할 수 도 있고 그때는 랜덤 액세스 이용