

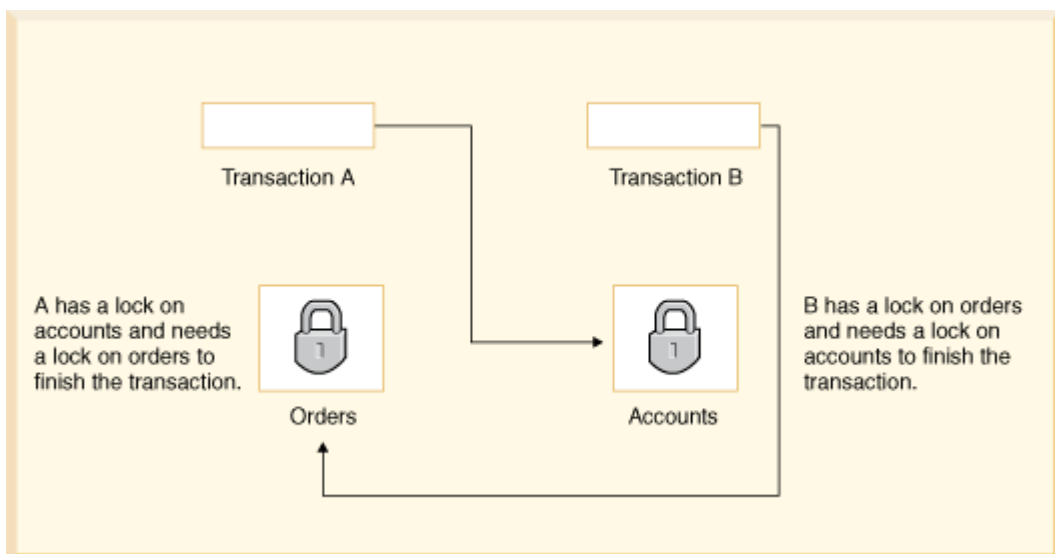
# 7. 그림으로 공부하는 오라클 구조 -

## Ch8. 오라클의 대기과 락

- 값을 변경하는 작업(UPDATE)을 할 때는 변경하기 전의 값을 바라봐도 상관없는 검색(SELECT)를 제외하고는 아무도 그 값에 접근할 수 없도록 해야 한다.

→ UPDATE 문은 이러한 이유로 자동으로 LOCK을 걸게 된다. 이 경우 **LOCK에 의한 대기(WAIT)**가 발생한다.

- Dead Lock : 서로가 상대가 보유하고 있는 LOCK을 기다리느라 교착 상태 발생



- 예시)
- Transaction A는 트랜잭션을 끝내기 위해 Accounts 를 변경하고 Orders를 변경해야 한다. (Order를 참조하여 회계전표를 생성하고 Order의 상태를 전기 상태로 바꾸기 위해)
- Transaction B는 Orders를 작업하고 Accounts를 작업해야 한다.(Accounts에 전기된 오더를 취소하기 위해 Accounts 테이블을 변경해야 할 때)
- 이 때, 마침 서로가 원하는 다음 데이터가 각각 상대방이 Lock을 걸고 있는 데이터일 때 Dead Lock이 발생하게 된다.

```
-- TRANSACTION A - ACCOUNTS의 NO = 1 ROW의 LOCK을 보유
UPDATE ACCOUNTS SET COL1 = 2 WHERE NO = 1
```

```
UPDATE ORDERS SET COL1 = 2 WHERE NO = 2
```

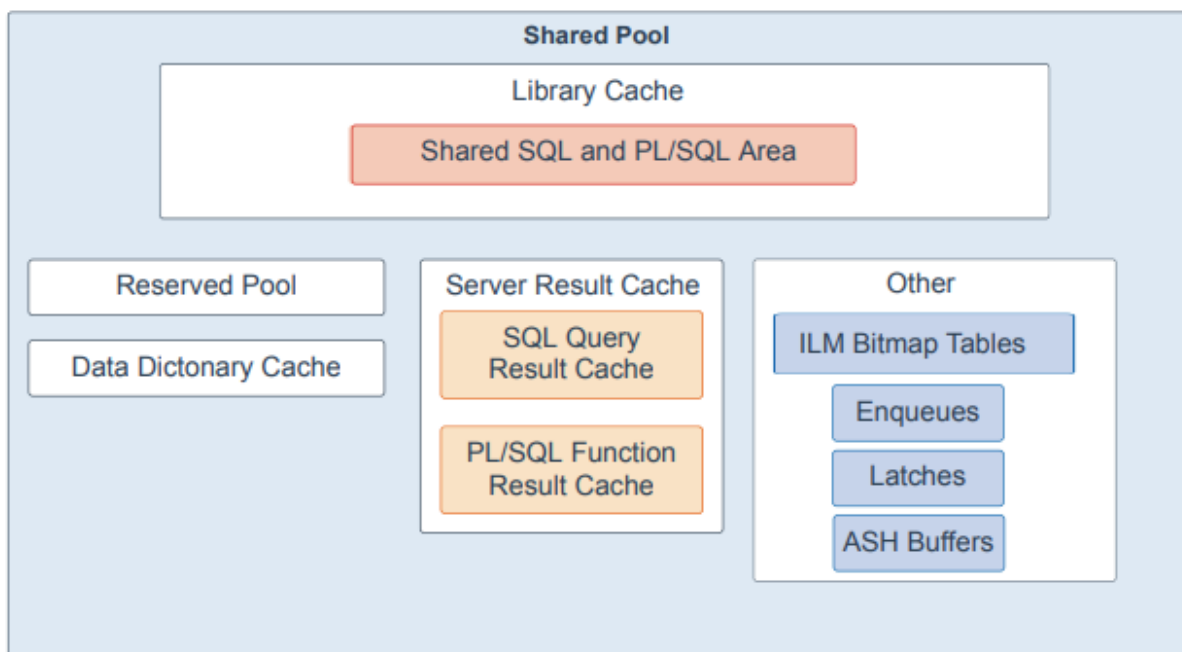
-- TRANSACTION B - ORDERS의 NO = 2 ROW의 LOCK을 보유

```
UPDATE ORDERS SET COL1 = 3 WHERE NO = 2
```

```
UPDATE ACCOUNTS SET COL1 = 3 WHERE NO = 1
```

→ DEAD LOCK이 발생하여 ORA-00060 에러코드 발생 → 오라클에 의해 한 쪽 ROLLBACK 후 처리

- LATCH : SGA(메모리공간) 내부의 공유 데이터에 대한 배타적인 LOCK 보장
  - SGA에 접근하는 프로세스는 원하는 영역에 대한 LATCH를 획득해야 접근할 수 있다.



- OS에 의한 선점 스케줄링(preemption) 발생 가능 (DEADLOCK 같은 것)

