**트랜잭션과 OWI**

- 트랜잭션 개요

사용자가 DML을 수행하게 되면 오라클은 내부적으로 다음과 같은 순서로 작업을 진행한다.

1) 해당 트랜잭션에 대해 언두 세그먼트를 할당한다. 할당이 실패하면 오프라인 상태의 언두 세그먼트를 온라인화해서 사용한다. 이 과정마저 실패하면, 새로운 언두 세그먼트를 생성한다.

2) 언두 세그먼트를 할당 받으면, 언두 세그먼트 헤더에 트랜잭션 테이블 슬롯을 생성한다.

3) 트랜잭션 테이블을 생성하고 나면 TXID를 생성하고, 현재 트랜잭션에 할당한다. 트랜잭션은 반드시 언두 영역을 할당 받은 다음에 ID를 부여받는다.

4) 트랜잭션의 대상이 되는 블록들을 버퍼 캐시로 적재하고 블록 헤더의 ITL에 트랜잭션 엔트리를 등록한다.

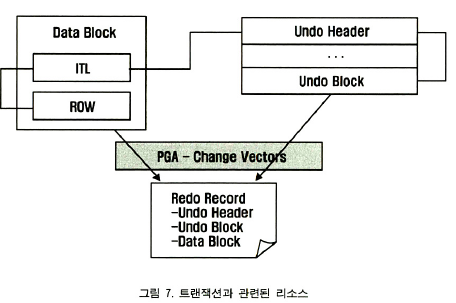
5) 변경할 블록들의 변경 정보는 PGA에 체인지 벡터라는 이름으로 저장한다. 보통 하나의 로우가 변경되는 경우 각각 언두 헤더 블록, 언두 블록, 데이터 블록에 해당하는 체인지 벡터들이 생긴다. 프로세스는 PGA의 체인지 벡터들을 리두 레코드라는 이름으로 리두 버퍼로 복사한다.

6) 이전 이미지에 대한 정보를 언두 블록에 기록하고, 데이터 블록을 변경한다. 변경된 데이터 블록은 더티 상태가 된다. 또한 변경된 데이터 블록에 대한 CR블록이 버퍼 캐시에 생성된다.

7) 커밋이 수행되면, 트랜잭션에 SCN을 할당한다. 커밋 정보는 리두 버퍼에 저장된다.

8) 언두 세그먼트 헤더의 트랜잭션 테이블에 커밋이 이루어졌음을 저장하고, 락을 포함한 모든 리소스에 대한 점유를 해제한다.

9) 리두 버퍼의 내용이 리두 로그 파일에 기록된다. 변경된 블록들은 이후에 DBWR에 의해 데이터 파일로 기록된다.



- 트랜잭션과 블록 덤프

트랜잭션에서 커밋이 이루어졌는지 아닌지의 여부는 Delayed Block Cleanout 때문에 실제로는 언두 세그먼트 헤더의 트랜잭션 테이블을 조회해보아야 알 수 있다. 오라클의 로우 레벨 락은 해당 [로우의 변경 여부 + 로우를 변경한 트랜잭션 정보 + 언두 영역의 테이블 슬롯 + TX 락]이 조합된 일종의 논리적인 락이다. TX락 자체가 로우 레벨 락을 의미하는 것은 아니며, 다양한 정보들이 조합되어 로우 레벨 락이 되는 것이다.

- 트랜잭션과 언두 헤더 덤프

