

ex) rownum 의 잘못된 사용 사례

WHERE ROWNUM = N ( N > 1 )

WHERE ROWNUM > N ( N > 1 )

▶ ROWNUM=1 은 ROWNUM<=1 로 사용하자

CURSOR\_SHARING 을 FORCE나 SIMILAR로 적용 하였을 경우  
ROWNUM=1을 ROWNUM=:\"SYS\_B\_0\"

1이라는 상수값을 바인드 변수값으로 재해석하여 실행한다

고로 전체 DATA를 다 수행한 뒤 1개 값을 가져오므로 성능악화 된다

하지만 ROWNUM<=:\"SYS\_B\_0\" 는 값보다 작거나 같은 값이라는 범위  
(STOPKEY)가 정해지기 때문에 변수의 값을 확인하여  
작거나 같은 데이터를 추출하므로 전체 데이터를 처리하지 않고, 1건만 추출하고  
SQL의 수행은 종료된다

▶ INDEX\_DESC와 ROWNUM<=1을 함께 사용하지 말자

완벽해 보이는 방법일수도 있겠으나 index가 unusabled이거나 없는 경우라면  
데이터의 정합성이 훼손될 수 있으므로 이 방법 대신

max() 혹은 order by desc 후 rownum <= 1 값을 가져오는 쿼리를 쓰는 방법이 좋다.

ROWNUM의 실행순서는 간단하게 표현해서

FROM > WHERE > ROWNUM > ORDER BY 정도 되겠다.

속도가 느린 경우는 TABLE FULL SCAN을 해버린 경우이다.

속도가 느리지 않은 경우는 TABLE FULL SCAN이 일어나지 않는 경우이다.

매번 실행할 때마다 옵티마이저가 실행계획을 세울텐데(이번 개발은 조희쿼리를 오라클 FUNCTION으로 만들어서 커서를 반환하는 형식이라 같은 쿼리가 매번 메모리로 올라가지 않는다) 그때마다 실행계획이 뒤죽박죽이 되는 케이스...

결국 TABLE FULL SCAN이 일어나는 원인은 WHERE가 없는 상태에서 ROWNUM을 사용하면, 순위를 메겨야 하기 때문에 기본적으로 오라클이 전체 TABLE FULL SCAN으로 데이터에 순위를 메겨주고, 그중에 한건을 갖고 오기 때문에 느린 것이다.