

EXISTS(서브 쿼리)는 서브 쿼리의 결과가 "**한 건이라도 존재하면**" **TRUE** 없으면 **FALSE**를 리턴한다.

EXISTS는 서브 쿼리에 일치하는 결과가 한 건이라도 있으면 쿼리를 더 이상 수행하지 않는다.

아래의 샘플 테이블을 참고하여 예제 쿼리를 보면 쉽게 이해할 수 있을 것이다.

EMP 테이블				DEPT_HISTORY 테이블			
EMPNO	ENAME	JOB	DEPTNO	EMPNO	SDATE	EDATE	DEPTNO
7698	BLAKE	MANAGER	30	7782	1981-06-09	1991-06-08	40
7782	CLARK	MANAGER	10	7566	1981-04-02	1991-04-01	10
7566	JONES	MANAGER	20	7566	1991-04-02	2001-04-01	30

```
SELECT a.empno
      , a.ename
      , a.deptno
FROM emp a
WHERE a.job = 'MANAGER'
      AND EXISTS (SELECT 1
                  FROM dept_history
                  WHERE aa.empno =
```

EXISTS, NOT EXISTS 사용법

```
SELECT a.empno
      , a.ename
      , a.deptno
FROM emp a
WHERE a.job = 'MANAGER'
      AND EXISTS (SELECT 1
                  FROM dept_history aa
                  WHERE aa.empno = a.empno)
```

EMPNO	ENAME	DEPTNO
7566	JONES	20
7782	CLARK	10

- dept_history 테이블에 JONES 2건, CLARK 1건, BLAKE 0건으로 BLAKE는 제외하고 조회 된다.
- 서브 쿼리의 dept_history 테이블에 데이터가 한 건이라도 존재해야 데이터가 조회 된다.
- [**SELECT 1**] SELECT 절에 컬럼은 불필요하기 때문에 의미없는 1을 기입 하였다.

IN과 EXISTS 비교

```
SELECT a.empno
      , a.ename
      , a.deptno
FROM emp a
WHERE a.job = 'MANAGER'
      AND a.empno IN (SELECT aa.empno
                      FROM dept_history aa
                      WHERE aa.empno = a.empno)
```

EMPNO	ENAME	DEPTNO
7566	JONES	20
7782	CLARK	10

- 위의 쿼리를 비교해 보면 IN과 EXISTS는 같은 결과가 조회된다.

- IN은 서브쿼리 결과를 모두 수행하고, EXISTS는 일치하는 결과가 있으면 더 이상 수행하지 않는다.

FROM절에
첨가된 테이블을
RAM에
올려놓는 것은
효율함.

- 서브 쿼리 테이블(dept_history)에 데이터량이 많으면 EXISTS를 사용하는것이 성능이 좋다.

1. 성능

-EXISTS: 조건에 해당하는 ROW의 존재 유무와 체크 후 더 이상 수행하지 않음.

SELECT 절을 평가하지 않으므로 일

SELECT 절을 사용하지 않음.

반적으로 IN에 비해 성능이 좋다.

-IN : 조건에 해당하는 ROW의 칼럼을 비교하여 체크한다.

SELECT 절에서 조회한 칼럼 값으로 비교하므로 EXISTS에 비해 성능이 떨어진다.

2. 실행 순서

-EXISTS: 메인 쿼리의 결과값을 서브 쿼리에 대입하여 조건 비교 후 결과를 출력한다. 메인쿼리->EXISTS쿼리

correlated subquery


-IN : 서브 쿼리의 결과값을 메인 쿼리에 대입하여 조건 비교 후 결과를 출력한다. IN쿼리->메인쿼리

2. EXISTS를 이용한 방식

```
SELECT *  
FROM tableA  
WHERE EXISTS (SELECT 1 FROM tableB WHERE title  
= 'TOPCREDU' AND tableA.id = tableB.id);
```

3. 조인을 이용한 방식

```
SELECT *  
FROM tableA  
JOIN tableB  
ON tableA.id = tableB.id  
WHERE tableB.title = 'TOPCREDU';
```

 *해당 서브쿼리에서 Table full scan이 발생하면, 전체적인 쿼리 속도가 늦어지게 될 수 있음.*

IN을 이용하게 되면 서브쿼리 전체를 먼저 스캔하고 tableA
의 모든 레코드에 대해
IN에서 추출된 데이터와 비교하게 되므로 서브쿼리의 추출
되는 레코드가 많다면 성능이 저하됩니다.

EXISTS와 조인을 이용하는 방법은 비교해서 참 또는 거짓만
리턴하므로 서브쿼리의
추출되는 데이터가 많다면 훨씬 좋은 성능을 보이게 됩니다.

이런저런 문제점들이 있었지만 제일 문제가 되는건 조건절에 쓴 IN 절이었다. IN은 ROW를 체크만하는 EXISTS랑 다르게 ROW의 데이터를 모두 확인하기 때문에 목적에 따라서는 EXISTS에 비해 훨씬 성능이 떨어질 수 있다는 점이다. 위 쿼리에선 굳이 IN을 안 쓰고도 단순히 TABLE3의 COL1이라는 값이 A 테이블의 COL1과 동일한 값들만 체크하면 되는 거라 EXISTS로도 대체 가능하였다.

```
SELECT A.COL1, A.COL2, A.COL3, B.COL1, B
FROM TABLE1 A JOIN TABLE2 B
ON A.PKID = B.PKID
WHERE EXISTS (SELECT 1
              FROM TABLE3 C
              WHERE C.COL2 = 'B'
              AND A.COL1 = C.COL1)
```

위와 같이 수정하였더니 5~6초에 걸리던 시간이 0. 몇 초대로 줄어들었다. 이렇게 함수 수정은 완료되었다.

결론

그래서 결론을 내리자면,

1. 엄청 많은 데이터를 비교하는 부분은 IN절보단 EXISTS를 이용할 수 있다면 되도록 EXISTS로 대체하자.
2. 조건절에는 가공될 수 있는(?), 바뀔 수 있는 서브쿼리 절은 넣으면 좋지 않다고 한다. 고정된 컬럼과 비교할 수 있는 조건절을 넣어주도록 하자.

하지만 EXISTS와 IN은 다른 점이 존재하므로 이에 대해 **유의**해야 한다.

EXISTS는 단지 ROW가 존재하는지만 체크하고 더 이상 수행 되지 않으나,

IN은 실제 존재하는 데이터들의 모든 값까지 확인한다.

따라서, EXISTS 연산자가 더 좋은 성능을 보이므로 가능하면 EXISTS를 사용하는 것이 바람직해 보인다.