

서브쿼리(Sub-Query)



배재대학교 컴퓨터공학과
김 창 수

Contents



서브쿼리의 개념



서브쿼리의 종류



서브쿼리

❖ 개요

- 하나의 SQL 명령문의 결과를 다른 SQL 명령문에 전달하기 위해 두 개 이상의 SQL 명령문을 하나의 SQL 명령문으로 연결하여 처리하는 방법
- 서브쿼리의 필요성
 - 예 : 교수 테이블에서 ‘전은지’ 교수와 직급이 동일한 모든 교수의 이름 검색
 - 처리방식
 1. 교수 테이블에서 ‘전은지’ 교수의 직급 검색 SQL 명령문 실행
 2. 교수 테이블의 직급 칼럼에서 1 에서 얻은 결과 값과 동일한 직급을 가진 교수 검색 명령문 실행
 - 1, 2 SQL 명령문 처리과정을 통합하여 하나의 SQL 명령문으로 처리

서브쿼리의 개념

❖ 개요

■ 서브쿼리 필요성 - 계속

서브쿼리를 이용한 교수명과 직급 검색

```
SELECT name, position  
FROM professor  
WHERE position
```

②

```
= (SELECT positon  
FROM professor  
WHERE name = '전은지' );
```

①

메인쿼리(main query)

서브쿼리(subquery)

■ 처리과정

1. 서브쿼리는 메인쿼리가 실행되기 전에 한번씩 실행됨
2. 서브쿼리에서 실행된 결과가 메인 쿼리에 전달되어 최종적인 결과를 출력

서브쿼리의 개념

서브쿼리를 사용하지 않은 경우

교수 테이블에서 '전은지' 교수의 직급 검색

```
SELECT position
FROM professor
WHERE name = '전은지' ;
```

①

PROFNO	NAME	...	POSITION	...
9901	김도훈	...	교수	...
9902	이재우	...	조교수	...
.....
9907	전은지		전임강사	

교수 테이블에서 '전임강사' 직급의 교수 검색

```
SELECT name, position
FROM professor
WHERE position = '전임강사' ;
```

②

③

NAME	POSITION
염일웅	전임강사
전은지	전임강사

서브쿼리를 사용한 경우

서브쿼리를 이용하여 교수 이름과 직급을 동시 검색

```
SELECT name, position
FROM professor
WHERE position = (SELECT position
FROM professor
WHERE name = '전은지' );
```

② ← 결과 전달

①

PROFNO	NAME	POSITION
9901	김도훈	교수
9902	이재우	조교수
.....
9904	염일웅	전임강사
.....
9907	전은지	전임강사

PROFNO	NAME	POSITION
9901	김도훈	교수
9902	이재우	조교수
.....
9904	염일웅	전임강사
.....
9907	전은지	전임강사

③

NAME	POSITION
염일웅	전임강사
전은지	전임강사



단일행 서브쿼리

❖ 개요

- 서브쿼리에서 단 하나의 행만을 검색하여 메인쿼리에 반환하는 질의문
- 메인쿼리의 WHERE 절에서 서브쿼리의 결과와 비교할 경우에는 반드시 단일행 비교 연산자 중 하나만 사용해야함
 - 단일 행 비교 연산자 : $=$, $>$, $>=$, $<$, $<>$, $<=$
- 서브쿼리의 결과로 하나의 행만이 출력되어야 함
 - 방법 1. 서브쿼리의 조건절에서 기본 키나 고유 키를 '=' 비교 하는 방식
 - 방법 2. 서브쿼리의 SELECT 절에서 전체 집합을 대상으로 그룹 함수 사용

'=' 연산자를 이용한 단일행 서브쿼리

❖ 사용 예

- 사용자 아이디가 'jun123'인 학생과 같은 학년인 학생의 학번, 이름, 학년을 출력하여라

```
SQL> SELECT studno, name, grade
2 FROM student
3 WHERE grade = (SELECT grade
4                 FROM student
5                 WHERE userid = 'jun123');
```

단일행 비교연산자(=)를 사용하기 위해서는,
서브쿼리의 결과가 반드시 단일행이어야한다

STUDNO	NAME	G
10101	전인하	4
10202	오유석	4
10107	이광훈	4

'<' 연산자를 이용한 단일행 서브쿼리

❖ 사용 예

- 101번 학과 학생들의 평균 몸무게보다 몸무게가 적은 학생의 이름, 학과번호, 몸무게를 출력하여라

```
SQL> SELECT name, deptno, weight
2 FROM student
3 WHERE weight < (SELECT AVG(weight)
4 FROM student
5 WHERE deptno = 101)
6 ORDER BY deptno;
```

101번 학과 학생의 평균 몸무게
(68kg) 는 단일값이다.

NAME	DEPTNO	WEIGHT
박미경	101	52
지이은	101	42
임은지	101	54
김지영	102	48
이노현	201	64
김지영	201	51
조명현	201	62

단일행 서브쿼리

❖ 실습 예

- 김사랑 사원과 부서가 같고, 급여는 김사랑 보다 많은 사원의 이름, 직급, 급여, 부서번호를 출력하여라

```
SQL> select ename, job, sal,deptno
  2   from emp
  3   where deptno = (select deptno
  4                     from emp
  5                     where ename = '김사랑')
  6   and   sal > (select sal
  7                  from emp
  8                  where ename = '김사랑');
```

ENAME	JOB	SAL	DEPTNO
이병헌	부장	600	20
안성기	사장	1000	20
박중훈	부장	560	20

다중 행 서브쿼리

❖ 개요

- 서브쿼리에서 반환되는 결과 행이 하나 이상일 때 사용하는 서브쿼리
- 메인쿼리의 WHERE 절에서 서브쿼리의 결과와 비교할 경우에는 다중 행 비교 연산자 를 사용하여 비교
 - 다중 행 비교 연산자 : IN, ANY, SOM, ALL, EXISTS
 - 다중 행 비교 연산자는 단일 행 비교 연산자와 결합하여 사용 가능

다중 행 비교 연산자

종류	의 미
IN	메인 쿼리의 비교 조건이 서브쿼리의 결과중에서 하나라도 일치하면 참, '= '비교만 가능
ANY, SOME	메인 쿼리의 비교 조건이 서브쿼리의 결과중에서 하나라도 일치하면 참, '= '비교만 가능
ALL	메인 쿼리의 비교 조건이 서브쿼리의 결과중에서 모든값이 일치하면 참,
EXISTS	메인 쿼리의 비교 조건이 서브쿼리의 결과중에서 만족하는 값이 하나라도 존재하면 참

IN 연산자를 이용한 다중 행 서브쿼리

❖ 개요

- 메인쿼리의 비교 조건에서 서브쿼리의 출력 결과와 하나라도 일치하면 메인쿼리 조건절이 참이 되는 연산자
- '=' 연산자를 OR로 연결한 것과 같은 의미
- IN 연산자를 이용한 서브쿼리의 예

```
SELECT name, grade, deptno
FROM student
WHERE deptno IN (SELECT deptno
FROM department
WHERE college = 100);
```

NAME	...	GRADE	DEPTNO
전인하	...	4	101
박미경	...	1	101
지은경	...	2	101
이광훈	...	4	101
...
오유석	...	4	102

=

```
SELECT name, grade, deptno
FROM student
WHERE deptno = 101
OR deptno = 102
```

IN 연산자를 이용한 다중 행 서브쿼리

❖ 사용 예

- 정보미디어학부(부서번호:100)에 소속된 모든 학생의 학번, 이름, 학과 번호를 출력하여라

```
SQL> SELECT name, grade, deptno
  2 FROM student
  3 WHERE deptno IN (SELECT deptno
  4 FROM department
  5 WHERE college = 100);
```

NAME	G	DEPTNO
전 이 하	4	101
박 미 이	1	101
지 이 진	2	101
서 재 진	1	101
류 미 정	2	101
이 가 오	4	101
이 영 아	2	101
이 영 아	3	101
이 영 아	2	102

ANY 연산자를 이용한 다중 행 서브쿼리

❖ 개요

- 메인쿼리의 비교 조건에서 서브쿼리의 출력 결과와 하나라도 일치하면 메인쿼리 조건절이 참이 되는 연산자
- '>,<' 등과 같은 범위 비교도 가능
- ANY 연산자를 이용한 서브쿼리의 예

```
SELECT studno, name, height
FROM student
WHERE height
```

```
> ANY (SELECT height
        FROM student
        WHERE grade = '4');
```

NAME	...	GRADE	HEIGHT
전인하	...	4	176
김영균	...	3	172
오유석	...	4	177
.....
이광훈	...	4	175
류민정	...	2	162

=

```
SELECT name, grade, height
FROM student
WHERE height > (SELECT MIN(height)
                 FROM student
                 WHERE grade = '4');
```

ANY

학생의 키가 176, 177,
175 중 최소인
175보다 크면 참

ANY 연산자를 이용한 다중 행 서브쿼리

❖ 사용 예

- 모든 학생 중에서 4학년 학생 중에서 키가 제일 작은 학생보다 키가 큰 학생의 학번, 이름, 키를 출력하여라

```
SQL> SELECT studno, name, height
2 FROM student
3 WHERE height > ANY (SELECT height
4 FROM student
5 WHERE grade = '4');
```

4학년의 최소 키인 175보다 키가 큰 학생을 검색하는 경우와 동일

STUDNO	NAME	HEIGHT
10101	전인하	176
20102	박동진	182
10202	오유석	177
10106	서재진	186
20104	조명훈	184



ALL 연산자를 이용한 다중 행 서브쿼리

❖ 개요

- 메인쿼리의 비교 조건에서 서브쿼리의 검색 결과와 모두 일치하면 메인쿼리 조건절이 참이 되는 연산자
- ANY 연산자와 차이점
 - ‘>ANY’ : 서브쿼리 결과 중에서 최소 값보다 크면 메인쿼리의 비교 조건이 참
 - ‘>ALL’ : 서브쿼리의 결과 중에서 최대 값보다 크면 메인쿼리의 비교 조건이 참

ALL 연산자를 이용한 다중 행 서브쿼리

```
SELECT studno, name, height
FROM student
WHERE height > ALL (SELECT height
```

```
FROM student
WHERE grade = '4');
```

=

```
SELECT name, grade, height
FROM student
WHERE height > (SELECT MAX(height)
FROM student
WHERE grade = '4');
```

NAME	...	GRADE	HEIGHT
전인하	...	4	176
김영균	...	3	172
오유석	...	4	177
.....
이광훈	...	4	175
류민정	...	2	162

ALL

학생의 키가 176,
177, 175 중 최대인
177보다 크면 참

ALL 연산자를 이용한 다중 행 서브쿼리

❖ 사용 예

- 모든 학생 중에서 4학년 학생 중에서 키가 가장 큰 학생보다 키가 큰 학생의 학번, 이름, 키를 출력하여라

```
SQL> SELECT studno, name, height
2 FROM student
3 WHERE height > ALL (SELECT height
4                     FROM student
5                     WHERE grade = '4');
```

STUDNO	NAME	HEIGHT
20102	박동진	182
10106	서재진	186
20104	조명훈	184

EXISTS 연산자를 이용한 다중 행 서브쿼리

❖ 개요

- 서브쿼리에서 검색된 결과가 하나라도 존재하면 메인쿼리 조건절이 참이 되는 연산자
- 서브쿼리에서 검색된 결과가 존재하지 않으면 메인쿼리의 조건절은 거짓
 - ‘선택된 레코드가 없습니다’ 라는 메시지 출력
- NOT EXISTS
 - EXISTS 와 상반되는 연산자
 - 서브쿼리에서 검색된 결과가 하나도 존재하지 않으면 메인쿼리의 조건절이 참이 되는 연산자

EXISTS 연산자를 이용한 다중 행 서브쿼리

❖ EXISTS 연산자를 이용한 다중 행 서브쿼리의 처리과정

```
SELECT profno, name, sal, comm, ..②
FROM professor
WHERE EXISTS (SELECT position
               FROM professor ①
               WHERE comm IS NOT NULL);
```

①의 결과가 존재하는가?

아니오
(①의 결과가 없는 경우)

예

(①의 결과가 한 행이라도 있는 경우)

②의 결과값은?

PROFNO	NAME	...	POSITION	COMM
9901	김도훈	...	교수	20
9902	이재우	...	조교수	
9903	성연희	...	조교수	15
.....
9907	전은지	...	전임강사	
9908	남은혁	...	부교수	17

②의 결과값은?

선택된 레코드가 없습니다

EXISTS 연산자를 이용한 다중 행 서브쿼리

❖ 사용 예

- 보직수당을 받는 교수가 한 명이라도 있으면 모든 교수의 교수 번호, 이름, 보직수당 그리고 급여와 보직수당의 합을 출력하여라

```
SQL> SELECT profno, name, sal, comm, SAL+NUL(comm, 0)
2 FROM professor
3 WHERE EXISTS (SELECT profno
4 FROM professor
5 WHERE comm IS NOT NULL);
```

PROFNO	NAME	SAL	COMM	SAL+NUL(COMM,0)
9901	김도훈	500	20	520
9902	이재우	320		320
9903	성영희	360	15	375
9904	정영민	240		240
9905	권영민	450	25	475
9906	이만식	420		420
9907	전이희	210		210
9908	남이희	400	17	417

NOT EXISTS 연산자를 이용한 다중 행 서브쿼리

❖ 실습 예

- 학생 중에서 'goodstudent'이라는 사용자 아이디가 없으면 1을 출력하여라

```
SQL> run
1 SELECT 1 userid_exist
2 FROM dual
3 WHERE NOT EXISTS (SELECT userid
4                   FROM student
5                   WHERE userid = 'goodstudent')

USERID_EXIST
-----
1
```

다중 컬럼 서브쿼리

❖ 개요

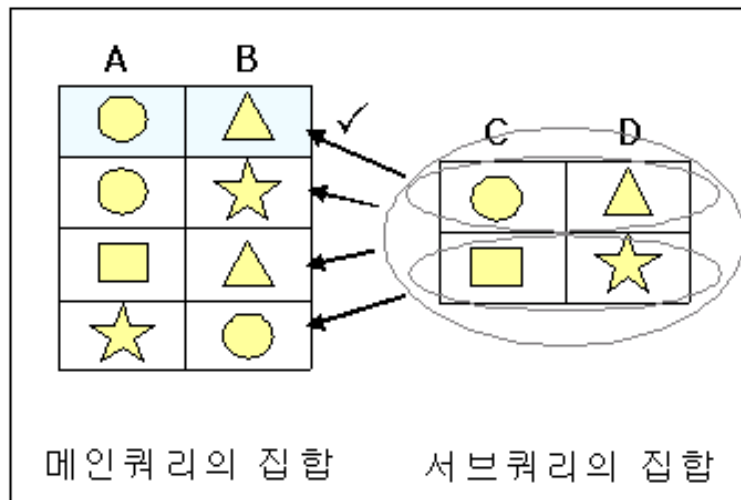
- 서브쿼리에서 여러 개의 컬럼 값을 검색하여 메인쿼리의 조건절과 비교하는 서브쿼리
- 메인쿼리의 조건절에서도 서브쿼리의 컬럼 수만큼 지정해야 함
- 종류
 - PAIRWISE : 컬럼을 쌍으로 묶어서 동시에 비교하는 방식
 - UNPAIRWISE : 컬럼별로 나누어서 비교한 후, AND 연산을 하는 방식

다중 컬럼 서브쿼리

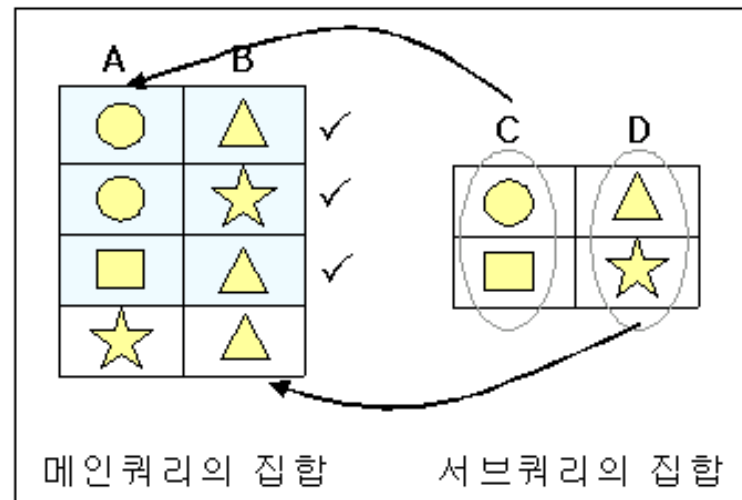
❖ 개요

▪ PAIRWISE, UNPAIRWISE 서브쿼리 비교

• PAIRWISE 비교 방법



• UNPAIRWISE 비교 방법



PAIRWISE 다중 칼럼 서브쿼리

❖ 개요

- 메인쿼리와 서브쿼리의 비교 대상 칼럼을 쌍으로 묶어서 행별로 비교하는 방법
- 메인쿼리와 서브쿼리에서 비교하는 칼럼의 수는 반드시 동일해야 함

❖ 사용법

```
SELECT    column_list
FROM      table1
WHERE      (column1, column2, ...) IN (SELECT    column1, column2, ...
                                       FROM      table2
                                       WHERE      condition);
```


PAIRWISE 다중 칼럼 서브쿼리

❖ 사용 예

- PAIRWISE 비교 방법에 의해 학년별로 몸무게가 최소인 학생의 이름, 학년, 몸무게를 출력하여라.

```
SQL> SELECT name, grade, weight
2 FROM student
3 WHERE (grade, weight) IN (SELECT grade, MIN(weight)
4 FROM student
5 GROUP BY grade);
```

NAME	G	WEIGHT
박미영	1	52
지이진	2	42
윤진하	3	70
전인하	4	72

UNPAIRWISE 다중 칼럼 서브쿼리

❖ 개요

- 메인쿼리와 서브쿼리의 비교 대상 칼럼을 분리하여 개별적으로 비교한 후 AND 연산에 의해 최종 결과를 출력
- 각 칼럼이 동시에 만족하지 않더라도 개별적으로 만족하는 경우에는 비교 조건이 참이 되어 결과를 출력 가능

❖ 사용법

```
SELECT    column_list
FROM      table1
WHERE      column1 IN (SELECT    column1
                        FROM      table2
                        WHERE      condition)
AND        column2 IN (SELECT    column2
                        FROM      table2
                        WHERE      condition) ;
```

UNPAIRWISE 다중 칼럼 서브쿼리

❖ 사용 예

- UNPAIRWISE 비교 방법에 의해 학년별로 몸무게가 최소인 학생의 이름, 학년, 몸무게를 출력하여라.

```
SQL> SELECT name, grade, weight
2 FROM student
3 WHERE grade IN (SELECT grade
4 FROM student
5 GROUP BY grade)
6 AND weight IN (SELECT MIN(weight)
7 FROM student
8 GROUP BY grade);
```

NAME	G	WEIGHT
지은경	2	42
박미경	1	52
박동진	1	70
윤진욱	3	70
서재진	1	72
전인하	4	72
류민정	2	72

학년이 1, 2, 3, 4종의 하나이고, 몸무게는 42, 52, 70, 72종의 하나와 같으면 참이다. PAIRWISE에서 출력이 안된 박동진, 서재진, 류민정 학생이 출력된다.

상호연관 서브쿼리

❖ 개요

- 메인쿼리절과 서브쿼리간에 검색 결과를 교환하는 서브쿼리
- 메인쿼리와 서브쿼리간의 결과를 교환하기 위하여 서브쿼리의 WHERE 조건절에서 메인쿼리의 테이블과 연결
- 주의
 - 행을 비교할 때마다 결과를 메인으로 반환하는 관계로 처리 성능이 저하될 수 있음

❖ 사용법

```
SELECT    column_list
FROM      table1
WHERE     [column | expression] operator
          (SELECT [column | expression]
FROM      table2
WHERE     table2.column operator table1.column) ;
```

상호연관 서브쿼리

❖ 사용 예

- 각 학과 학생의 평균 키보다 키가 큰 학생의 이름, 학과 번호, 키를 출력하여라

```
SQL> SELECT  name, deptno, height
  2 FROM      student s1
  3 WHERE     height > (SELECT  AVG(height)
  4                      FROM    student s2
  5                      WHERE    s2.deptno = s1.deptno)
  6 ORDER BY deptno;
```

NAME	DEPTNO	HEIGHT
전인하	101	176
서재진	101	186
이관훈	101	175
오유선	102	177
이진욱	102	171
박동진	201	182
조명훈	201	184

1. 메인쿼리에서 학생 테이블의 학과 번호를 읽어서 서브쿼리로 전달
2. 메인쿼리에서 전달받은 학과 번호로 서브쿼리에서 학과의 평균 키를 계산
3. 메인쿼리에서 해당학과의 평균 키보다 큰 학생의 이름, 학과번호, 키를 출력

실무에서 서브쿼리 사용시 주의사항

❖ 단일행 서브쿼리에서 오류가 발생하는 경우

- 복수 행 값을 반환하는 서브쿼리와 단일행 비교연산자 함께 사용하는 경우
- 반환되는 칼럼의 수와 메인쿼리에서 비교되는 칼럼 수가 일치하지 않는 경우
- 복수행을 출력하는 서브쿼리와 '=' 단일행 연산자로 비교하는 경우

```
SQL> SELECT name, grade, weight  
2 FROM student  
3 WHERE weight = (SELECT MIN(weight)  
4 FROM student  
5 GROUP BY grade);
```

```
WHERE weight = (SELECT MIN(weight)  
*)
```

3행에 오류:

ORA-01427: 단일 행 부속 질의에 2개 이상의 행이 리턴되었습니다

grade의 종류가 4개이므로
서브쿼리의 결과 행은 4개

실무에서 서브쿼리 사용시 주의사항

❖ 메인쿼리와 서브쿼리 컬럼의 수가 일치하지 않는 경우

❖ 사용 예

- 101번 학과 교수 중에서 최소 급여를 받는 교수의 이름, 직급, 급여를 출력하여라.

```
SQL> SELECT name, position, sal
  2 FROM professor
  3 WHERE sal = (SELECT MIN(sal), comm
  4                FROM professor
  5                WHERE deptno = 101);
WHERE sal = (SELECT MIN(sal), comm
              *
```

3행에 오류:

ORA-00913: 값의 수가 너무 많습니다

컬럼의 수가 일치하지 않는 경우

```
SQL> SELECT name, position, sal
  2 FROM professor
  3 WHERE sal = (SELECT MIN(sal)
  4                FROM professor
  5                WHERE deptno = 101);
```

NAME	POSITION	SAL
전은지	전임강사	210

컬럼의 수가 일치하는 경우

실무에서 서브쿼리 사용시 주의사항

❖ ORDER BY 절 사용

- 서브쿼리 내에서 ORDER BY 절 사용하면 오류 발생

```
SQL> SELECT  profno, name
  2 FROM      professor
  3 WHERE     sal > ANY(SELECT  sal
  4                                FROM      professor
  5                                WHERE     position = '조교수')
  6 ORDER BY profno);
ORDER BY profno)
*
```

6행에 오류:

ORA-00933: SQL 명령어가 올바르게 종료되지 않았습니다

```
SQL> SELECT  profno, name
  2 FROM      professor
  3 WHERE     sal > ANY(SELECT  sal
  4                                FROM      professor
  5                                WHERE     position = '조교수')
  6 ORDER BY profno;
```

PROFNO NAME

PROFNO	NAME
9901	김도훈
9903	성연희
9905	권현일
9906	이만식
9908	남지현

실무에서 서브쿼리 사용시 주의사항

❖ 서브쿼리의 결과가 NULL인 경우

❖ 사용 예

- 2002년에 입사한 교수의 평균 급여보다 많은 급여를 받는 교수의 교수 번호, 이름, 급여를 출력하여라.

```
SQL> SELECT  profno, name, sal
  2 FROM    professor
  3 WHERE   sal > (SELECT  AVG(sal)
  4                FROM    professor
  5                WHERE   TO_CHAR(hiredate, 'YYYY') = '2002');
```

선택된 레코드가 없습니다.

```
SQL> SELECT  AVG(sal)
  2 FROM    professor
  3 WHERE   TO_CHAR(hiredate, 'YYYY') = '2002';
```

AVG(SAL)

결과 값이 NULL