

[Database] Subquery(단일,다중) 사용법

노트북: [TIL-MY]

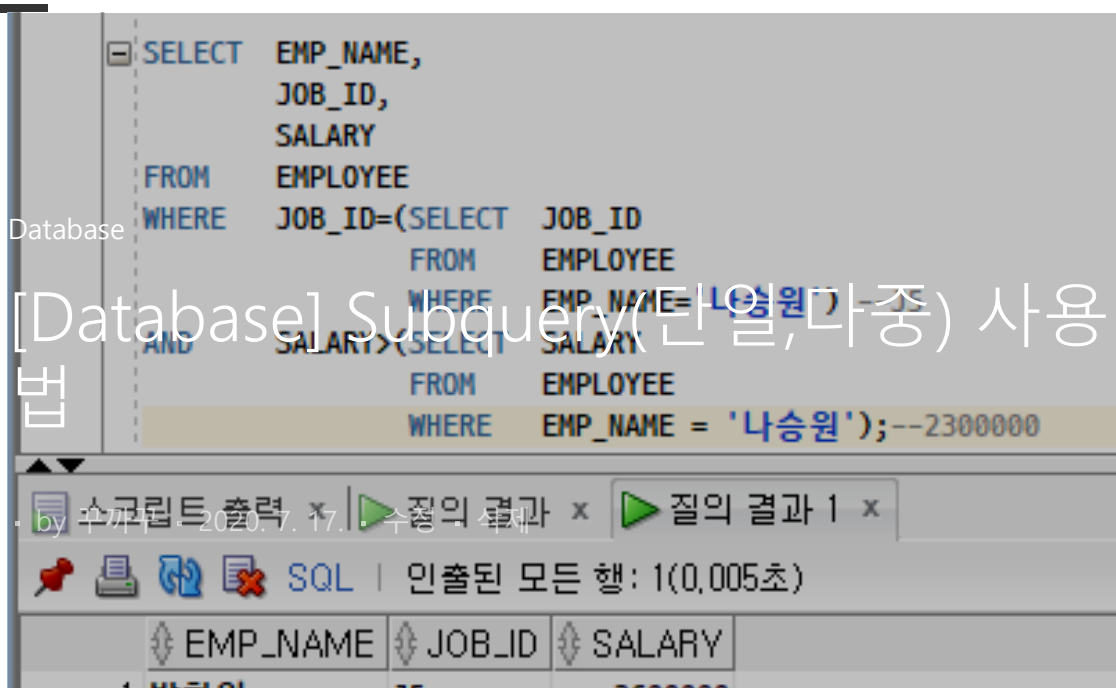
만든 날짜: 2020-07-17 오전 9:01

URL: <https://continuous-development.tistory.com/27>

나무늘보의 개발 블로그

홈

태그



Subquery

서브 쿼리란 하나의 쿼리가 다른 쿼리에 포함되는 구조로서

분류 전체보기

Python

Database

ASP.NET

Algorithm

Deep learning

다른 쿼리에 포함된 내부 쿼리는 외부 쿼리에 사용
될 값을 반환하는 역할을 한다.

서브쿼리는 메인쿼리에게 데이터를 넘겨주는 것 이
다.

SELECT

FROM

WHERE expr operator(SELECT ...

FROM

...

WHERE

...) ;

식의 구문으로 이루어진다.

Subquery의 유형은 총 4개다

단일(행) - 단일열 / 다중열

다중(행) - 단일열 / 다중열

으로 나뉜다. 여기서 열은 행열 할때 열이다. 단일 다
중은 행을 나타낸다.

단일 행일 경우 비교 연산자(=, >, >=, <=, <>)를 사
용한다.

AWS

ETC..

공지사항

글 보실 때 주의사
항

: 최근글 : 인
기글

[Data
Sub...



2020.07.17

[Data
SEP...



2020.07.17

[Data
Self...



2020.07.17

[Data
IN...



2020.07.16

[Data
OR...



2020.07.16

최근댓글

태그

사용법,

ORDER BY 사용,

OUTER JOIN 함

수,

다중일 경우에는 (IN, ANY,ALL)을 사용한다.

Subquery - 단일 행 서브쿼리

서브쿼리에 대한 결과 값이 단일 행 (단 하나의 행)으로 나오는 쿼리

The screenshot shows a SQL IDE with a query window and a results window. The query is as follows:

```
-- Subquery 단일 행 서브쿼리
SELECT EMP_NAME,
       JOB_ID,
       SALARY
FROM   EMPLOYEE
WHERE  JOB_ID=(SELECT JOB_ID
                 FROM   EMPLOYEE
                 WHERE  EMP_NAME='나승원') --J5
AND    SALARY>(SELECT SALARY
                 FROM   EMPLOYEE
                 WHERE  EMP_NAME = '나승원');--2300000
```

The results window shows the following data:

EMP_NAME	JOB_ID	SALARY
1 박하일	J5	2600000

IN Operator,
날짜함수,
JOIN 함수,
Oracle SQL,
INNER JOIN 함수,
Subquery 함수,
인스턴스,
selfjoin,
서브쿼리 사용법,
설정, AWS,
self join 사용법,
ASNI,
selfjoin사용법,
SUBQUERY사용,
INNER JOIN
OUTER JOIN,
UNION함수,
SEP Operator,
SQL,
UNION ALL함수,
GROUP BY 사용,
INTERSECT함수,
MINUS함수,
self join 함수,
Oracle,
SUBQUERY함수,
selfjoin함수

전체 방문자

88

Today : 0

Yesterday : 3

```
SELECT EMP_NAME,  
       JOB_ID,  
       SALARY  
FROM   EMPLOYEE  
WHERE  SALARY > ( SELECT AVG(SALARY)  
                  FROM   EMPLOYEE);
```

Subquery - 다중 행 서브쿼리

다중 서브 쿼리 앞에는 보통 IN / ANY / ON 를 사용한다.

기본적인 비교 연산자를 사용 할 수 없는 이유는 하나의 값이 아니기 때문이다.

그래서 IN / ANY / ON 추가로 사용한다.

NOT IN 에서는 문법상 서브쿼리에서 WHERE조건 절을 통해서 NULL이 아닌 조건만 가져와야 한다.

IN 연산자

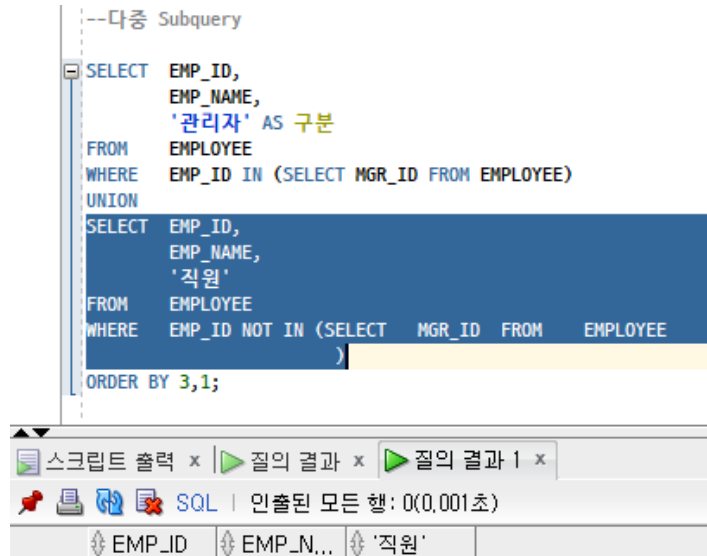
```
--다중 Subquery
SELECT EMP_ID,
       EMP_NAME,
       '관리자' AS 구분
FROM   EMPLOYEE
WHERE  EMP_ID IN (SELECT MGR_ID FROM EMPLOYEE)
UNION
SELECT EMP_ID,
       EMP_NAME,
       '직원'
FROM   EMPLOYEE
WHERE  EMP_ID NOT IN (SELECT MGR_ID FROM EMPLOYEE
                      WHERE MGR_ID IS NOT NULL)
ORDER BY 3,1;
```

스크립트 출력 x | 질의 결과 x | 질의 결과 1 x

SQL | 인출된 모든 행: 22(0.003초)

	EMP_ID	EMP_NAME	구분
1	100	한선기	관리자
2	101	강중훈	관리자
3	104	안석규	관리자
4	141	김예수	관리자
5	174	전우성	관리자
6	200	고승우	관리자
7	102	최만식	직원
8	103	정도연	직원
9	107	조재형	직원
10	124	정지현	직원
11	143	나승원	직원
12	144	김순이	직원
13	149	성해교	직원
14	176	엄정하	직원
15	178	신성규	직원

여기서 주의 할점은 NOT IN은 서브쿼리에서 하나라도 널이 있으면 널로 출력한다.



이 UNION을 CASE WHEN 으로 나타내면

SELECT	EMP_ID,	
	EMP_NAME,	
	'관리자' AS 구분	
FROM	EMPLOYEE	
WHERE	EMP_ID IN (SELECT MGR_ID FROM EMPLOYEE)	
UNION		
SELECT	EMP_ID,	
	EMP_NAME,	
	'직원'	
FROM	EMPLOYEE	
WHERE	EMP_ID NOT IN (SELECT MGR_ID FROM EMPLOYEE	
	WHERE MGR_ID IS NOT NULL)	
ORDER BY	3,1;	
--CASE WHEN으로 나타낸 쿼리		
SELECT	EMP_ID,	
	EMP_NAME,	
	CASE WHEN EMP_ID IN (SELECT MGR_ID	
	FROM EMPLOYEE) THEN '관리자'	
	WHEN EMP_ID NOT IN (SELECT MGR_ID	
	FROM EMPLOYEE	
	WHERE MGR_ID IS NOT NULL) THEN '직원'	
	END AS 구분	
FROM	EMPLOYEE	
ORDER BY	3,1;	
SELECT	EMP_ID,	
	EMP_NAME,	
	CASE WHEN EMP_ID IN (SELECT MGR_ID	
	FROM EMPLOYEE) THEN '관리자'	
	ELSE '직원'	
	END AS 구분	
FROM	EMPLOYEE	
ORDER BY	3,1;	

스크립트 출력 x	질의 결과 x	질의 결과 1 x
SQL 인출된 모든 행: 22(0,002초)		
EMP_ID	EMP_NAME	구분
1 100	한선기	관리자
2 101	강중훈	관리자
3 104	안석규	관리자
4 141	김예수	관리자
5 174	전우성	관리자
6 200	고승우	관리자
7 102	최만식	직원

ANY 연산자

< ANY : 비교 대상 중 최대 값 보다 작음

> ANY : 비교 대상 중 최소 값 보다 큼

= ANY : IN 연산자와 동일

```
-- ANY
SELECT EMP_NAME,
       SALARY
FROM   EMPLOYEE
JOIN   JOB USING(JOB_ID)
WHERE  JOB_TITLE = '대리'
AND    SALARY > ANY
      (SELECT SALARY
       FROM   EMPLOYEE
       JOIN   JOB USING (JOB_ID)
       WHERE  JOB_TITLE = '과장');
```

질의 결과 x

SQL | 인출된 모든 행: 3(0.005초)

	EMP_NAME	SALARY
1	엄정하	2420000
2	권상후	3410000
3	임영애	2640000

```
SELECT EMP_NAME,
       SALARY
FROM   EMPLOYEE
JOIN   JOB USING(JOB_ID)
WHERE  JOB_TITLE = '대리'
AND    SALARY < ANY
      (SELECT SALARY
       FROM   EMPLOYEE
       JOIN   JOB USING (JOB_ID)
       WHERE  JOB_TITLE = '과장');
```

질의 결과 x

SQL | 인출된 모든 행: 2(0.003초)

	EMP_NAME	SALARY
1	전우성	2090000
2	엄정하	2420000

ALL 연산자는

< ALL: 비교 대상 중 최소 값 보다도 작음

> ALL: 비교 대상 중 최대 값 보다도 큼

= ALL: IN 연산자와 동일


```
--ALL 연산자
SELECT EMP_NAME,
       SALARY
FROM   EMPLOYEE
JOIN   JOB USING(JOB_ID)
WHERE  JOB_TITLE = '대리'
AND    SALARY > ALL
      (SELECT SALARY
       FROM   EMPLOYEE
       JOIN   JOB USING (JOB_ID)
       WHERE  JOB_TITLE = '과장');
```

질의 결과 x

SQL | 인출된 모든 행: 2(0.001초)

	EMP_NAME	SALARY
1	권상후	3410000
2	임영애	2640000

```
SELECT EMP_NAME,
       SALARY
FROM   EMPLOYEE
JOIN   JOB USING(JOB_ID)
WHERE  JOB_TITLE = '대리'
AND    SALARY < ALL
      (SELECT SALARY
       FROM   EMPLOYEE
       JOIN   JOB USING (JOB_ID)
       WHERE  JOB_TITLE = '과장');
```

질의 결과 x

SQL | 인출된 모든 행: 2(0.001초)

	EMP_NAME	SALARY
1	권상후	3410000
2	임영애	2640000

다중행에 다중열 서브쿼리

--다중행에 다중열 서브쿼리

SELECT

EMP_NAME,

JOB_TITLE,

SALARY

FROM

EMPLOYEE

JOIN

JOB USING(JOB_ID)

WHERE

(JOB_ID,SALARY) IN (

SELECT

JOB_ID,TRUNC(AVG(SALARY),-5)

FROM

EMPLOYEE

GROUP BY

JOB_ID);

--이 WHERE 절에 조건도 맞춰줘야 한다.서브쿼리에서 SELECT 의 갯수가 여러개면(다중열)이면 WHERE에서

--받는 갯수도 맞춰서 같아줘야 한다.

질의 결과

SOL

인출된 모든 행: 6(0.004초)

	EMP_NAME	JOB_TITLE	SALARY
1	한선기	대표이사	9000000
2	안석규	부장	3500000
3	나승원	과장	2300000
4	김술오	차장	2500000
5	이종기	차장	2500000
6	김우섭	차장	2500000

다중열일 경우는 where 에서도 다중으로 받아줘야 한다.



'Database' 카테고리의 다른 글

[Database] Subquery(단일,다중) 사용법 (0)	08:59:05
[Database] SEP Operator(UNION,UNION ALL, INTERSECT,MINUS) & IN Operator사용법 (0)	02:42:43
[Database] Self JOIN 사용법 (ANSI 표준) (0)	01:53:21
[Database] INNER JOIN / OUTER JOIN 함수 사용법 및 차이(ORACLE & ANSI표준) (0)	2020.07.16
[Database] ORDER BY 절,GROUP BY 절 함수 사용법 (Oracle SQL) (0)	2020.07.16
[Database] DECODE & CASE 함수 사용법 (Oracle SQL) (0)	2020.07.15

태그

- Oracle SQL
- subquery
- Subquery 함수
- SUBQUERY사용
- SUBQUERY함수
- 다중 행
- 단일 행
- 서브쿼리
- 서브쿼리 사용법

관련글

스프링트 분석 > 결과 결과 >

SQL 인출된 모든 행: 25(0.01초)

EMP_ID	ROLE_NAME
1 100	SALES
2 101	SALES
3 102	SALES
4 103	SE
5 104	SE
6 105	SE

EMP_NAME
EMP_NO
EMAIL
PHONE
HIRE_DATE
JOB_ID (FK)
SALARY
BONUS_PCT
MARRIAGE
MARRIAGE_ID (FK)

WHERE E.DEPT_ID = D.DEPT_ID

스프링트 분석 > 결과 결과 >

SQL 인출된 모든 행: 20(0.002초)

EMP_NAME	DEPT_NAME
1 한민기	재무경영팀
2 강동준	재무경영팀
3 최한석	재무경영팀
4 김도현	기술개발팀
5 김민준	기술개발팀

OR DEPT_ID IS NULL
(ORDER BY SALARY DESC)

스프링트 분석 > 결과 결과 >

SQL 인출된 모든 행: 9(0.002초)

EMP_NAME	SALARY
1 김승현	3000000
2 박희정	2600000
3 이승훈	2300000
4 신하균	2300000

[Database]... [Database]... [Database]... [Database]...

댓글 0