

## [Database] SET Operator(UNION, UNION ALL, INTERSECT, MINUS) & IN Operator 사용법

노트북: [TIL-MY]

만든 날짜: 2020-07-17 오전 9:01

URL: <https://continuous-development.tistory.com/26>

# 나무늘보의 개발 블로그

홈

태그

Database	스크립트 출력 x	▶ 질의 결과 x
[Database] SQL   인출된 모든 행: 25(0.01초)		
	EMPID	ROLENAME
1	100	SALES
2	101	SALES
3	102	SALES
4	103	SE
5	104	SE

by 꾸까꾸 · 2020. 7. 17. · 수정 · 삭제

## SET Operator

--구문  
SELECT

분류 전체보기 

Python

Database 

ASP.NET

Algorithm

Deep learning

FROM  
UNION / UNION ALL / INTERSECT / EXCEPT  
SELECT  
FROM  
- 개념

두개 이상의 쿼리 결과를 하나로 결합시키는 연산자

SELECT 에 들어가는 컬럼의 개수나 타입이 값이 같  
아야 된다.

## UNION

결과에 대한 합집합 / 모든 중복된 행은 하나의 행으  
로만든다.

AWS

ETC..

공지사항

글 보실 때 주의사  
항

: 최근글 : 인  
기글

[Data  
Sub...



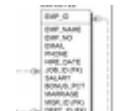
2020.07.17

[Data  
SEP...



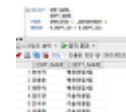
2020.07.17

[Data  
Self...



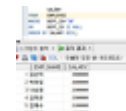
2020.07.17

[Data  
IN...



2020.07.16

[Data  
OR...



2020.07.16

최근댓글

태그

사용법,

ORDER BY 사용,

OUTER JOIN 함

수,

```
--SET OPERATOR
```

```
SELECT EMP_ID,
        ROLE_NAME
FROM   EMPLOYEE_ROLE
UNION
SELECT EMP_ID,
        ROLE_NAME
FROM   ROLE_HISTORY;
```

	EMP_ID	ROLE_NAME
1	100	SALES
2	101	SALES
3	102	SALES
4	103	SE
5	104	SE
6	104	SE-ANLY
7	104	SE-PKG
8	107	SE
9	124	MKT
10	141	MKT
11	143	MKT
12	144	MKT
13	149	MKT

UNION

## UNION ALL

결과에 대한 합집합 / 모든 중복된 행도 표시

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

```
--UNION ALL
SELECT EMP_ID,
       ROLE_NAME
FROM   EMPLOYEE_ROLE
UNION ALL
SELECT EMP_ID,
       ROLE_NAME
FROM   ROLE_HISTORY;
```

스크립트 출력 x    실행 결과 x

SQL | 인출된 모든 행: 26(0,001초)

	EMP_ID	ROLE_NAME
1	100	SALES
2	101	SALES
3	102	SALES
4	103	SE
5	104	SE
6	107	SE
7	124	MKT
8	141	MKT
9	143	MKT
10	144	MKT
11	149	MKT
12	174	SALES-MKT
13	176	SALES-MKT
14	178	(null)

UNION ALL

## INTERSECT

결과에 대한 교집합

```
--INTERSECT
SELECT EMP_ID,
        ROLE_NAME
FROM EMPLOYEE_ROLE
INTERSECT
SELECT EMP_ID,
        ROLE_NAME
FROM ROLE_HISTORY;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.001초)

	EMP_ID	ROLE_NAME
1	104	SE

INTERSECT

## MINUS

결과에 대한 차집합

```
--MINUS
SELECT EMP_ID,
        ROLE_NAME
FROM EMPLOYEE_ROLE
MINUS
SELECT EMP_ID,
        ROLE_NAME
FROM ROLE_HISTORY;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 21(0.1초)

	EMP_ID	ROLE_NAME
3	102	SALES
4	103	SE
5	107	SE
6	124	MKT
7	141	MKT
8	143	MKT
9	144	MKT
10	149	MKT
11	174	SALES-MKT
12	176	SALES-MKT
13	178	(null)
14	200	HR
15	201	MKT

MINUS

만약에 컬럼수를 맞추기 힘들다면 이런식으로 더미  
(임시)컬럼을 넣을 수도 있다.

--컬럼 맞추기

```

SELECT EMP_NAME,
       JOB_ID,
       HIRE_DATE
FROM   EMPLOYEE
WHERE  DEPT_ID = '20'
UNION
SELECT DEPT_NAME,
       DEPT_ID,
       NULL
FROM   DEPARTMENT
WHERE  DEPT_ID = '20'

```

스크립트 출력 x | 질의 결과 x | 질의 결과 1 x

SQL | 인출된 모든 행: 4(0.002초)

	EMP_NAME	JOB_ID	HIRE_DATE
1	감우섭	J4	05/07/31
2	김술오	J4	96/10/01
3	이중기	J4	04/10/01
4	회계팀	20	(null)

```

SELECT EMP_NAME,
       SALARY
FROM   EMPLOYEE
WHERE  DEPT_ID = '20'
UNION
SELECT DEPT_NAME,
       DEPT_ID
FROM   DEPARTMENT
WHERE  DEPT_ID = '20';

```

스크립트 출력 x | 질의 결과 x | 질의 결과 1 x

SQL | 실행 중: SELECT EMP\_NAME, SALARY FROM EMPLOYEE

ORA-01790: expression must have same datatype as corresponding expression  
 01790. 00000 - "expression must have same datatype as corresponding expression"  
 \*Cause:  
 \*Action:  
 206행, 9열에서 오류 발생

지금 이거 같은경우에 컬럼의 갯수는 맞지만 DEPT\_ID는 문자열인 반면에 SALARY는 숫자형이다. 서로 타입이 달라 오류가 나고 있다.

```
SELECT EMP_NAME,  
       SALARY  
FROM   EMPLOYEE  
WHERE  DEPT_ID = '20'  
UNION  
SELECT DEPT_NAME,  
       TO_NUMBER(DEPT_ID)  
FROM   DEPARTMENT  
WHERE  DEPT_ID = '20';
```

스크립트 출력 x | 질의 결과 x | 질의 결과 1 x

SQL | 인출된 모든 행: 4(0.001초)

	EMP_NAME	SALARY
1	감우섭	2500000
2	김술오	2500000
3	이중기	2500000
4	회계팀	20

이걸 TO\_NUMBER로 형태를 바꿔 사용한다.

```

SELECT EMP_ID,
       EMP_NAME,
       '관리자' AS 구분
FROM EMPLOYEE
WHERE EMP_ID = '141'
AND DEPT_ID = '50'
UNION
SELECT EMP_ID,
       EMP_NAME,
       '직원' AS 구분
FROM EMPLOYEE
WHERE DEPT_ID='50'
AND EMP_ID !='141'
ORDER BY 3;

```

스크립트 출력 x | 질의 결과 x | 질의 결과 1 x

SQL | 인출된 모든 행: 6(0,004초)

	EMP_ID	EMP_NAME	구분
1	141	김예수	관리자
2	124	정지현	직원
3	143	나승원	직원
4	144	김순이	직원
5	149	성해교	직원
6	201	박하일	직원

이런식으로 IF문처럼 쓸수도 있다. 그 결과 값을 합쳐서 나오게 한다.

3번째 컬럼은 리터널을 써서 WHERE 조건에 맞는 값들에 대해 컬럼에 대한 속성값으로 넣고 그 두개의 테이블을 합치는 쿼리다.

## IN OPERART



SELECT

FROM

WHERE 컬럼 IN (")

IN 을 WHERE 절에 넣어줘서 두가지 경우에 대한 WHERE 절을 만들어 넣어준다.

이 결과 값과 밑에 유니온을 해서 나온 결과 값이 같다.

```
--IN OPERATOR

SELECT EMP_NAME,
       JOB_TITLE 직급
FROM   EMPLOYEE
JOIN   JOB USING (JOB_ID)
WHERE  JOB_TITLE IN ('대리', '사원')
ORDER BY 2,1;

SELECT EMP_NAME,
       '사원' 직급
FROM   EMPLOYEE
JOIN   JOB USING (JOB_ID)
WHERE  JOB_TITLE = '사원'
UNION
SELECT EMP_NAME,
       '대리'
FROM   EMPLOYEE
JOIN   JOB USING (JOB_ID)
WHERE  JOB_TITLE = '대리'
ORDER BY 2,1;
```

스크립트 출력 x | 질의 결과 x | 질의 결과 1 x

SQL | 인출된 모든 행: 8(0.002초)

	EMP_NAME	직급
1	권상후	대리
2	엄정하	대리
3	임영애	대리
4	전우성	대리
5	고승우	사원
6	성해교	사원
7	엄정하	사원
8	정지현	사원

[Database] SEP Operator(UNION,UNION ALL, INTERSECT,MINUS) & IN Operator사용법



'Database' 카테고리의 다른 글

[Database] Subquery(단일,다중) 사용법 (0)

08:59:05

[Database] SEP Operator(UNION,UNION ALL, INTERSECT,MINUS) & IN

02:42:43

## Operator사용법 (0)

[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

IN Operator

INTERSECT

INTERSECT함수

Minus

MINUS함수

SEP Operator


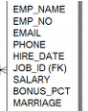
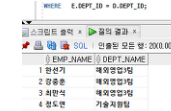

Union

Union All

UNION ALL함수

UNION함수

## 관련글

			
---	---	---	---

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

## 댓글 0

TEL. 02.1234.5678 / 경기 성남시 분당구 판교역로

© Kakao Corp.

