

2021-05-27 목요일
ORACLE

2021.05.27, ORACLE, 1.JOIN 함수-표준

<설명> JOIN함수는 앞서 말했듯이 공통된 특성을 찾아 테이블을 이어 새로운 테이블을 만드는 함수이다

<예시> JOIN함수의 형식에는 CROSS JOIN, INNER JOIN, LEFT JOIN, RIGHT JOIN, SELF JOIN, FULL JOIN이 있다.

이 중 INNER 조인이 대표적으로 사용되는데 형식은 **SELECT 요소 FROM TABLE1 INNER JOIN TABLE2 ON 조건**의 형식을 갖추어야 사용할 수 있다. 그래서 이 조건을 KEY로 설정해서 이어주는 방식이다. 가령 employee의 ename의 첫번째 글자가 department의 loc의 첫번째 글자와 같을 때 그 부분 옆에 LOC를 출력하라는 함수는 아래와 같다.

```
SELECT * FROM employee A INNER JOIN department B ON SUBSTR(A.ename,1,1) = SUBSTR(B.loc,1,1);
```

BLAKE의 예를 보면 ename의 첫번째 B와 loc의 B가 일치하므로 이어준다는 함수를 통해 테이블이 합쳐진 것을 확인할 수 있다.

	ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO	DNO_1	DNAME	LOC
1	7698	BLAKE	MANAGER	7839	81/05/01	2850	(null)	30	40	OPERATIONS	BOSTON
2	7782	CLARK	MANAGER	7839	81/06/09	2450	(null)	10	30	SALES	CHICAGO

<문제> **EMPLOYEE**의 ename과 **EMPLOYEES**의 first이 같을 경우 phone_number를 추가하는 테이블을 만들고 싶다.

해당하는 쿼리문을 작성하시오. (ename, first_name, phone_number 출력)

HINT : EMPLOYEES에서 FIRST_NAME은 대소문자가 합쳐져있다. 소문자를 대문자로 변경하는 함수는 UPPER(컬럼명) 이다.

	ENAME	FIRST_NAME	PHONE_NUMBER
1	JAMES	James	650.124.1334
2	JAMES	James	650.124.7234

2021.05.27, ORACLE, 2.JOIN 함수-Self JOIN 함수

<설명> SELF JOIN은 자기 자신의 테이블의 번호 등을 재 정의하여 마치 새로운 테이블을 붙이는 듯한 효과를 보는 함수이다.

<예시>

```
SELECT A.employee_id, A.first_name, A.manager_id, B.employee_id, B.first_name FROM employees A
INNER JOIN employees B ON A.manager_id=B.employee_id ;
```

출력한 결과를 보면 똑같은 EMPLOYEES의 employee_id를 출력했지만 A에서는 사원번호 B에서는 매니저번호가 employee_id로 등록된 것을 확인할 수 있다. 이는 ON조건에서 A.manger_id = B.employee_id로 묶었기 때문에 이렇게 출력된 것이다.

	EMPLOYEE_ID	FIRST_NAME	MANAGER_ID	EMPLOYEE_ID_1	FIRST_NAME_1
1	173	Sundita	148	148	Gerald
2	172	Elizabeth	148	148	Gerald
3	171	William	148	148	Gerald
4	170	Taylor	148	148	Gerald
5	169	Harrison	148	148	Gerald
6	168	Lisa	148	148	Gerald
7	103	Alexander	102	102	Lex

<문제> 위의 EMPLOYEES를 SELF JOIN 한 것에서 각 manager가 담당하는 사원이 몇명인지 파악하여라.

(HINT : 위의 SELF JOIN한 값에서 GROUP BY를 이용하여라.)

	EMPLOYEE_ID	FIRST_NAME	매니저의 담당 사원 수
1	114	Den	5
2	123	Shanta	8
3	121	Adam	8
4	201	Michael	1
5	146	Karen	6
6	147	Alberto	6
7	124	Kevin	8
8	145	John	6
9	102	Lex	1
10	108	Nancy	5
11	120	Matthew	8
12	101	Neena	5
13	148	Gerald	6
14	205	Shelley	1
15	103	Alexander	4
16	122	Payam	8
17	100	Steven	14
18	149	Eleni	6

2021.05.27, ORACLE, 3.JOIN 함수-비표준

<설명> 비표준 조인은 FROM절에 같이 받아와 WHERE절로 조건을 만드는 형식이다.

<예시> SELECT * FROM TABLE1, TABLE2 WHERE TABLE1.key = TABLE2.key;

SELECT ename,job,dname,loc FROM employee A,department B;

만 하게되면 Cartesian 곱에 따라서 모든 경우의수가 생성되는데 이를 아래와 같은 WHERE절로 필터링하여 원하는 테이블을 얻는다.

SELECT ename,job,dname,loc FROM employee A,department B WHERE A.dno = B.dno;

	ENAME	JOB	DNAME	LOC
1	SMITH	CLERK	RESEARCH	DALLAS
2	ALLEN	SALESMAN	SALES	CHICAGO
3	WARD	SALESMAN	SALES	CHICAGO
4	JONES	MANAGER	RESEARCH	DALLAS
5	MARTIN	SALESMAN	SALES	CHICAGO
6	BLAKE	MANAGER	SALES	CHICAGO
7	CLARK	MANAGER	ACCOUNTING	NEW YORK
8	SCOTT	ANALYST	RESEARCH	DALLAS
9	KING	PRESIDENT	ACCOUNTING	NEW YORK
10	TURNER	SALESMAN	SALES	CHICAGO
11	ADAMS	CLERK	RESEARCH	DALLAS
12	JAMES	CLERK	SALES	CHICAGO
13	FORD	ANALYST	RESEARCH	DALLAS
14	MILLER	CLERK	ACCOUNTING	NEW YORK

<문제> 아래 쿼리문은 INNER JOIN으로 설계한 테이블이다. 비표준으로 설계하여 아래와 같은 값을 얻으시오.

SELECT A.eno,A.ename,B.dname,A.dno FROM employee A INNER JOIN department B ON A.dno = B.dno AND A.eno = 7788;

	ENO	ENAME	DNAME	DNO
1	7788	SCOTT	RESEARCH	20

2021.05.27, ORACLE, 4.JOIN 함수응용 -더블 JOIN함수

<설명> INNER JOIN을 2번 사용하여 테이블을 2번 붙일 수 있다.

<예시> INNER JOIN을 2번 사용하면서 다른 한번은 조건으로서 사용하는 경우

아래 쿼리문의 salgrade는 ON 조건에 사잇 값이라는 조건을 주어 테이블을 이어주었다.

employee의 salary가 salgrade의 losal과 hisal 사이에 있으면 해당 부분을 KEY로 받아 이어지는 형태이다.

```
SELECT * FROM employee A INNER JOIN department B ON A.dno = B.dno INNER JOIN salgrade C ON A.salary BETWEEN C.losal AND C.hisal;
```

ENO	ENAME	JOB	MANAGER	HIREDATE	SALARY	COMMISSION	DNO	DNO_1	DNAME	LOC	GRADE	LOSAL	HISAL
1	7839 KING	PRESIDENT	(null)	81/11/17	5000	(null)	10	10	ACCOUNTING	NEW YORK	5	3001	9999
2	7902 FORD	ANALYST	7566	81/12/03	3000	(null)	20	20	RESEARCH	DALLAS	4	2001	3000
3	7788 SCOTT	ANALYST	7566	87/07/13	3000	(null)	20	20	RESEARCH	DALLAS	4	2001	3000
4	7566 JONES	MANAGER	7839	81/04/02	2975	(null)	20	20	RESEARCH	DALLAS	4	2001	3000
5	7698 BLAKE	MANAGER	7839	81/05/01	2850	(null)	30	30	SALES	CHICAGO	4	2001	3000
6	7782 CLARK	MANAGER	7839	81/06/09	2450	(null)	10	10	ACCOUNTING	NEW YORK	4	2001	3000
7	7499 ALLEN	SALESMAN	7698	81/02/20	1600	300	30	30	SALES	CHICAGO	3	1401	2000
8	7844 TURNER	SALESMAN	7698	81/09/08	1500	0	30	30	SALES	CHICAGO	3	1401	2000
9	7934 MILLER	CLERK	7782	82/01/23	1300	(null)	10	10	ACCOUNTING	NEW YORK	2	1201	1400
10	7521 WARD	SALESMAN	7698	81/02/22	1250	500	30	30	SALES	CHICAGO	2	1201	1400
11	7654 MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	30	SALES	CHICAGO	2	1201	1400
12	7876 ADAMS	CLERK	7788	87/07/13	1100	(null)	20	20	RESEARCH	DALLAS	1	700	1200
13	7900 JAMES	CLERK	7698	81/12/03	950	(null)	30	30	SALES	CHICAGO	1	700	1200
14	7369 SMITH	CLERK	7902	80/12/17	800	(null)	20	20	RESEARCH	DALLAS	1	700	1200

<문제> 위의 테이블에서 grade가 3이상인 사람의 eno, ename, dname, loc, grade를 출력하면서 grade가 가장 큰 것부터 나열하여라.

(HINT : 위의 TABLE을 FROM으로 받으면 쉽게 해결할 수 있을 것이다.)

...	ENAME	DNAME	LOC	GRADE
1	7839 KING	ACCOUNTING	NEW YORK	5
2	7782 CLARK	ACCOUNTING	NEW YORK	4
3	7566 JONES	RESEARCH	DALLAS	4
4	7698 BLAKE	SALES	CHICAGO	4
5	7902 FORD	RESEARCH	DALLAS	4
6	7788 SCOTT	RESEARCH	DALLAS	4