

231128 (sql,outerjoin,rightouterjoin)

자바스프링

2023/11/28 20:26

<http://blog.naver.com/sophia2164/223277366247>

다음 SQL 중 결과가 다른 SQL은 무엇인가?

- ① SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;
- ② SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 BETWEEN B.COL1 AND B.COL2;
- ③ SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL1 <= A.COL1
AND B.COL2 >= A.COL1;
- ④ SELECT B.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;

다음 SQL 중 결과가 다른 SQL은 무엇인가?

- ① SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;
- ② SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 BETWEEN B.COL1 AND B.COL2;
- ③ SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL1 <= A.COL1
AND B.COL2 >= A.COL1;
- ④ SELECT B.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;

다음 SQL 중 결과가 다른 SQL은 무엇인가?

- ① SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;
- ② SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 BETWEEN B.COL1 AND B.COL2;
- ③ SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL1 <= A.COL1
AND B.COL2 >= A.COL1;
- ④ SELECT B.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 >= B.COL1
AND A.COL1 <= B.COL2;

의미상 같다

```

1 SELECT A.PRODUCT_NAME,
2       B.MEMBER_ID,
3       B.CONTENT,
4       C.EVENT_NAME
5 FROM PRODUCT A,
6       PRODUCT_REVIEW B,
7       EVENT C
8 WHERE A.PRODUCT_CODE = B.PRODUCT_CODE
9       AND B.REG_DATE BETWEEN C.START_DATE AND C.END_DATE;

```

B.REG_DATE 는 C.start_date와 c.end_date 사이에 있다.(범위지정)

| PRODUCT_NAME | MEMBER_ID | CONTENT | EVENT_NAME |
|--------------|------------|--------------------|-------------|
| 무소용 무선 마우스 | sqlchild02 | 무선이라 정말 편하네요! | 20% 할인쿠폰 증정 |
| 무소용 무선 마우스 | sqlbaby01 | 무소용인데 소음이 조금 있는 듯? | 20% 할인쿠폰 증정 |
| 기계식 게이밍 키보드 | sqladult03 | 게이밍할 맛 납니다 | 마우스패드 증정 |

친구나 동료에게 Whiteboard를 추천할 의향이 있나요?

A.COL1 >= B.COL1
A.COL1 <= B.COL2

다음 SQL 중 결과가 다른 SQL은 무엇인가?

① SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

② SELECT A.COL1, B.COL2
FROM SAMPLE1 A INNER JOIN SAMPLE2 B
ON A.COL1 = B.COL1;

③ SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1(+); Outer join

④ SELECT B.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

S1
table A

| | | |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

S2
table B

| | | |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

[SQL]
SELECT SUM(A.COL2) AS RESULT1,
COUNT(A.COL3) AS RESULT2,
COUNT(DISTINCT A.COL3) AS RESULT3
FROM SAMPLE A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

[SAMPLE1 테이블]

| COL1 | COL2 | COL3 |
|------|------|------|
| 가 | 100 | A |
| 나 | 110 | A |
| 다 | 200 | NULL |
| 라 | 150 | B |
| 마 | 50 | NULL |

[SAMPLE2 테이블]

| COL1 | COL2 |
|------|------|
| 가 | 10 |
| 나 | 0 |
| 다 | 5 |

| Result1 | Result2 | Result3 |
|---------|---------|---------|
| 410 | 2 | 1 |

→ 3/0/1

친구나 동료에게 Whiteboard
할 의향이 있나요?

다음 두 테이블을 COL1을 JOIN 컬럼으로 하여 LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN했을 때 각각 출력되는 데이터 건수로 가장 적절한 것은 무엇인가? (단, SAMPLE1 테이블이 LEFT TABLE, SAMPLE2 테이블이 RIGHT TABLE이라고 가정)

(A) 3건

| COL1 | COL2 | COL3 |
|------|------|------|
| 1 | A | D |
| 3 | B | E |
| 5 | C | F |

2 =
A =
F =

| COL1 | COL2 |
|------|------|
| 2 | G |
| 3 | H |
| 3 | I |

6 A

| C1 | C2 | C3 | C1 | C4 |
|------|------|------|------|------|
| 1 | A | D | NULL | NULL |
| 3 | B | E | 3 | H |
| 3 | B | E | 3 | I |
| 5 | C | F | NULL | NULL |
| NULL | NULL | NULL | 2 | G |
| NULL | NULL | NULL | 6 | 12 |

[SAMPLE1 테이블]

| COL1 | COL2 |
|------|------|
| 2 | G |
| 3 | H |
| 3 | I |

[SAMPLE2 테이블]

| COL1 | COL2 |
|------|------|
| 1 | A |
| 2 | B |
| 3 | NULL |
| 4 | D |
| 5 | E |

1 =

5 | E | NULL | NULL

| COL1 | COL2 | COL1-1 | COL2-1 |
|------|------|--------|--------|
| 2 | G | 2 | B |

| COL1 | COL2 | COL1 | COL2 |
|------|------|------|------|
| NULL | NULL | 1 | A |
| 2 | G | 2 | B |
| NULL | NULL | 3 | NULL |
| NULL | NULL | 4 | D |
| NULL | NULL | 5 | E |

[SQL]

SELECT *

FROM SAMPLE1 A RIGHT OUTER JOIN SAMPLE2 B

ON A.COL1 = B.COL1

WHERE B.COL2 IS NOT NULL;

[SAMPLE1 테이블]

| COL1 | COL2 |
|------|------|
| 2 | G |
| 3 | H |
| 3 | I |

[SAMPLE2 테이블]

| COL1 | COL2 |
|------|------|
| 1 | A |
| 2 | B |
| 3 | NULL |
| 4 | D |
| 5 | E |

Select

from

outer join

on

where

group by
having
order by

| | | | |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| N | N | / | A |
| 2 | G | 2 | B |
| 3 | H | 3 | N |
| 3 | I | 3 | N |
| N | N | 4 | D |
| N | N | 5 | E |

다음 SQL 중 결과가 다른 SQL은 무엇인가?

① SELECT A.COL1, B.COL2

FROM SAMPLE1 A, SAMPLE2 B

WHERE A.COL1 = B.COL1;

② SELECT A.COL1, B.COL2

FROM SAMPLE1 A INNER JOIN SAMPLE2 B

ON A.COL1 = B.COL1;

③ SELECT A.COL1, B.COL2

FROM SAMPLE1 A, SAMPLE2 B

WHERE A.COL1 = B.COL1(+);

④ SELECT B.COL1, B.COL2

FROM SAMPLE1 A, SAMPLE2 B

WHERE A.COL1 = B.COL1;

다음 SQL 중 결과가 다른 SQL은 무엇인가?

① SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

② SELECT A.COL1, B.COL2
FROM SAMPLE1 A INNER JOIN SAMPLE2 B
ON A.COL1 = B.COL1;

③ SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1(+); Outer join

④ SELECT B.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

S1
table A

| | | |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

S2
table B

| | | |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

[SQL]
SELECT SUM(A.COL2) AS RESULT1,
COUNT(A.COL3) AS RESULT2,
COUNT(DISTINCT A.COL3) AS RESULT3
FROM SAMPLE A, SAMPLE2 B
WHERE A.COL1 = B.COL1;

[SAMPLE1 테이블]

| COL1 | COL2 | COL3 |
|------|------|------|
| 가 | 100 | A |
| 나 | 110 | A |
| 다 | 200 | NULL |
| 라 | 150 | B |
| 마 | 50 | NULL |

[SAMPLE2 테이블]

| COL1 | COL2 |
|------|------|
| 가 | 10 |
| 나 | 0 |
| 다 | 5 |

| Result1 | Result2 | Result3 |
|---------|---------|---------|
| 410 | 2 | 1 |

→ 3/0

친구나 동료에게 Whiteboard
할 의향이 있나요?

다음 SQL의 결과로 가장 적절한 것은 무엇인가?

[SQL]

```
SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL3 % 5 = 0;
```

[SAMPLE1 테이블]

| COL1 | COL2 |
|------|------|
| 2 | G |
| 4 | I |

[SAMPLE2 테이블]

| COL1 | COL3 |
|------|---------------|
| 1 | 55 |
| 2 | 12 |
| 3 | 30 |
| 4 | 33 |
| 4 | 20 |

| | A | B | C | D | E | F |
|----|----|----|----|--------------|---------------|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | c1 | c2 | c1 | c2 | |
| 4 | 1 | 2 | g | 1 | 55 | |
| 5 | 2 | 2 | g | 2 | 12 | |
| 6 | 3 | 2 | g | 3 | 30 | |
| 7 | 4 | 2 | g | 4 | 33 | |
| 8 | 5 | 2 | g | 5 | 20 | |
| 9 | 6 | 4 | i | 1 | 55 | |
| 10 | 7 | 4 | i | 2 | 12 | |
| 11 | 8 | 4 | i | 3 | 30 | |
| 12 | 9 | 4 | i | 4 | 33 | |
| 13 | 10 | 4 | i | 5 | 20 | |
| 14 | | | | | | |
| 15 | | | | | | |
| 16 | | | | | | |

1) 다음 SQL의 결과로 가장 적절한 것은 무엇인가?

[SQL]

```
SELECT A.COL1, B.COL3
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL3 % 5 = 0;
```

[SAMPLE1 테이블]

| COL1 | COL2 |
|------|------|
| 2 | G |
| 4 | I |

[SAMPLE2 테이블]

| COL1 | COL3 |
|------|------|
| 1 | 55 |
| 2 | 12 |
| 3 | 30 |
| 4 | 33 |
| 4 | 20 |

2) 다음 SQL 중 에러가 발생하지 않는 SQL은 무엇인가? (Oracle이라고 가정)

- ① SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE B.COL2 IS NULL;
- ② SELECT A.COL1, B.COL2
FROM SAMPLE1 A NATURAL JOIN SAMPLE2 B
ON A.COL1 = B.COL1;
- ③ SELECT A.COL1, B.COL2
FROM SAMPLE1 A, SAMPLE2 B
WHERE A.COL1(+) = B.COL1(+);
- ④ SELECT B.COL1, B.COL2
FROM SAMPLE1 A JOIN SAMPLE2 B
USING (COL1, COL2);

| 주요지목록 - Excel | | | | | | | | | | | | | | | |
|---------------|-----|----------|------|-----------------|---------------|--------------|------|------|------|------|------|------|------|------|------|
| 국악 | | | | | | | | | | | | | | | |
| 번호 | 성명 | 주민번호 | 전화번호 | 주소 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 | 전화번호 |
| 1001 | 심나래 | 920902-2 | 1001 | 서울특별시 관악구 청룡동 | 010-8947-7354 | 063-225-5975 | | | | | | | | | |
| 1002 | 최은배 | 970502-1 | 1002 | 경기도 광명시 | 010-1212-1212 | | | | | | | | | | |
| 1003 | 김은진 | 911030-2 | 1003 | 경기도 화성시 봉담읍 | 010-1321-1311 | | | | | | | | | | |
| 1004 | 박태석 | 980916-1 | 1004 | 서울특별시 관악구 신림동 | 010-3982-1757 | 02-1111-2222 | | | | | | | | | |
| 1005 | 박지영 | 970826-2 | 1005 | 서울특별시 관악구 | 010-0123-4567 | 02-000-1111 | | | | | | | | | |
| 1006 | 김정민 | 030505-4 | 1006 | 서울특별시 관악구 | 010-1111-3333 | 02-1234-1111 | | | | | | | | | |
| 1007 | 주원호 | 990710-1 | 1007 | 경기도 안양시 동안구 관왕동 | 010-1111-1111 | | | | | | | | | | |
| 1008 | 구경현 | 920806-2 | 1008 | 서울시 양천구 목동 | 010-4201-1111 | | | | | | | | | | |
| 1009 | 김민현 | 000101-3 | 1009 | 서울시 영등포구 시골대로 | 010-3999-5736 | | | | | | | | | | |
| 1010 | 윤승현 | 921024-1 | 1010 | 서울 관악구 봉천동 | 010-4154-4487 | | | | | | | | | | |
| 1011 | 손윤재 | 980605-1 | 1011 | 서울시 관악구 인현동 | 010-2009-3966 | 02-134-2345 | | | | | | | | | |
| 1012 | 김은진 | 980626-1 | 1012 | 서울시 서대문구 남가좌동 | 010-2717-3904 | 02-392-3202 | | | | | | | | | |
| 1013 | 오승현 | 801105-1 | 1013 | 경기도 고양시 일산구 대우동 | 010-8980-4801 | 031-903-0000 | | | | | | | | | |
| 1014 | 김영진 | 980227-1 | 1014 | 서울시 관악구 봉천동 | 010-4567-8927 | 061-535-2757 | | | | | | | | | |
| 1015 | 박태민 | 990909-1 | 1015 | 서울시 영등포구 대림동 | 010-1111-1111 | 0-2-222-2222 | | | | | | | | | |

다음 SQL 중 에러가 발생하지 않는 SQL은 무엇인가? (Oracle이라고 가정)

① SELECT A.COL1, B.COL2

FROM SAMPLE1 A, SAMPLE2 B

WHERE B.COL2 IS NULL;

② SELECT A.COL1, B.COL2

FROM SAMPLE1 A NATURAL JOIN SAMPLE2 B

ON A.COL1 = B.COL1;

③ SELECT A.COL1, B.COL2

FROM SAMPLE1 A, SAMPLE2 B

WHERE A.COL1(+) = B.COL1(+);

④ SELECT B.COL1, B.COL2

FROM SAMPLE1 A JOIN SAMPLE2 B

USING (COL1, COL2);

outer join

LEFT

(+)는 둘 중 1개만 쓴다
왼쪽에 쓰면 right outer join
오른쪽에 쓰면 left outer join

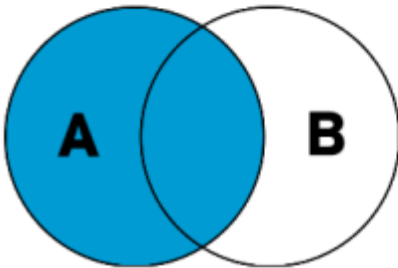
Right

특정 테이블 지정 사용 가능

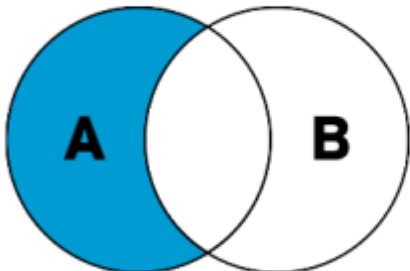
조금 더 알아보기

OUTER JOIN

LEFT OUTER JOIN

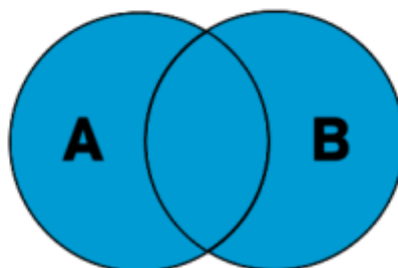


```
SELECT *  
FROM A a  
LEFT JOIN B b  
ON a.KEY = b.KEY
```

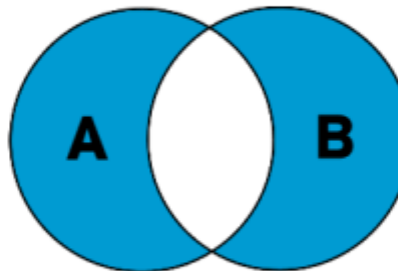


```
SELECT *  
FROM A a  
LEFT JOIN B b  
ON a.KEY = b.KEY  
WHERE b.KEY IS NULL
```

FULL OUTER JOIN

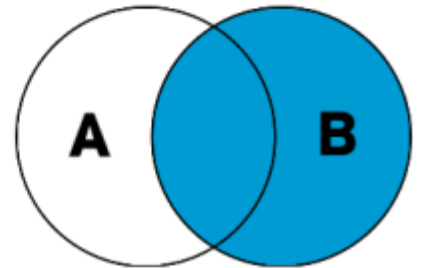


```
SELECT *  
FROM A a  
FULL OUTER JOIN B b  
ON a.KEY = b.KEY
```

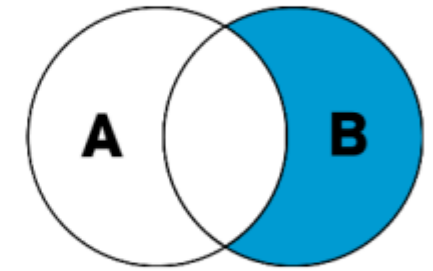


```
SELECT *  
FROM A a  
FULL OUTER JOIN B b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL  
OR b.KEY IS NULL
```

RIGHT OUTER JOIN



```
SELECT *  
FROM A a  
RIGHT OUTER JOIN B b  
ON a.KEY = b.KEY
```

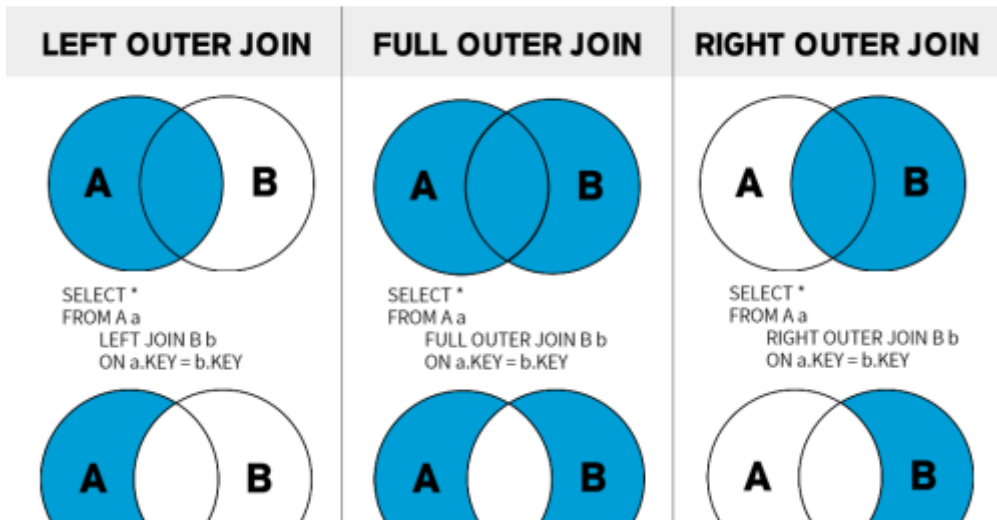


```
SELECT *  
FROM A a  
RIGHT OUTER JOIN B b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL
```

출처: <https://hongong.hanbit.co.kr/sql->

[%EA%B8%B0%EB%B3%B8-%EB%AC%B8%EB%B2%95-joininner-outer-cross-self-join/](https://hongong.hanbit.co.kr/sql-%EA%B8%B0%EB%B3%B8-%EB%AC%B8%EB%B2%95-joininner-outer-cross-self-join/)

SQL



SQL 기본 문법: JOIN(INNER, OUTER, CROSS, SELF JOIN)

조인은 두 개의 테이블을 서로 묶어서 하나의 결과를 만들어 내는 것을 말한다. INNER JOIN(내부 조인)은 두 테이블을 조인할 때, 두 테이블에 모두 지정한 열의 데이터가 있어야 한다. OUTER JOIN(외부 조인)은 두 테이블을 조인할 때...

hongong.hanbit.co.kr

