

```

1 REM Author :
2 REM Date :
3 REM Objective : Chapter 5. Join
4 REM Environment : Ubuntu Server 20.04 LTS, HeidiSQL 10.2.0, MySQL Community Server 5.7.34.0
5
6 REM JOIN(JOIN)
7 1. 한개 이상의 테이블로부터 데이터를 조회하는 것
8 2. 주로 Primary-Key와 Foreign-Key의 관계를 가진 컬럼을 소유하고 있는 테이블을 통한 검색 시 사용
9
10
11 REM CROSS JOIN
12 1. Cartesian Product(카티시안 곱)
13 2. 조인 조건이 부적합하거나 조인조건을 완전히 생략한 경우 행의 모든 조합을 표시
14 3. 첫번째 테이블의 모든 행이 두번째 테이블의 모든 행에 조인됨
15 4. 너무 많은 행을 생성하기 때문에 결과를 사용할 수 없다.
16 5. 모든 행을 조합해야 하는 경우가 아니라면 반드시 WHERE절에 적합한 조인 조건을 포함시켜야
17
18 SELECT empno, ename, dname
19 FROM emp, dept;
20
21 SELECT empno, ename, dname
22 FROM emp CROSS JOIN dept;
23
24
25
26 REM NATURAL JOIN
27 1. EQUI Join (등가조인), SIMPLE JOIN(단순조인), INNER JOIN(내부조인)
28 2. 2개 이상의 테이블이 공통되는 컬럼에 의해 논리적으로 결합되는 조인기법
29 3. WHERE절에 사용된 공통된 컬럼들이 동등 연산자(=)에 의해 비교
30
31 SELECT empno, ename, dname
32 FROM emp, dept
33 WHERE emp.deptno = dept.deptno;
34
35 --사원이름 KING 의 부서이름과 근무지를 출력하시오.
36 SELECT empno, ename, dname
37 FROM emp, dept
38 WHERE emp.deptno = dept.deptno AND ename = 'KING';
39
40
41 4. 테이블 별칭 사용
42 1)별칭의 Guideline
43 2)열 이름을 지정하는 경우 시간이 많이 걸리며 테이블 이름이 길 때는 더욱 오래 걸림
44 3)테이블 이름 대신 테이블 별칭을 사용가능
45 4)SQL 코드를 적게 작성해도 되므로 메모리 사용이 줄어듦
46 5)FROM 절에서 특정 테이블이름에 대해 별칭을 사용했다면 SELECT 문에서도 테이블 이름을 대신한다.
47 6)의미있는 이름을 주자.
48 7)별칭은 현재 SELECT에서만 유효하다.
49
50 SELECT e.empno, e.ename, d.dname
51 FROM emp e, dept d
52 WHERE e.deptno = d.deptno;
53
54 SELECT empno, ename, dname
55 FROM emp NATURAL JOIN dept;
56
57 SELECT empno, ename, job, dname, loc
58 FROM emp NATURAL JOIN dept
59 WHERE empno = 7900;
60
61
62
63 REM JOIN ~ USING
64 1. USING을 사용하지 않았다면 MySQL Server는 각 테이블에 공통된 컬럼을 자동으로 검색하여 비교
65 2. USING을 사용하면 USING절에 정의된 컬럼을 기준으로 Natural 조인이 발생한다.
66
67 SELECT empno, ename, dname
68 FROM emp JOIN dept USING (deptno)
69 WHERE empno = 7900;
70
71
72
73 REM JOIN ~ ON
74 1. EQUI JOIN에서는 WHERE절에서 작성했던 조인 조건을 ON절에서 작성한다.

```

```

75
76 SELECT empno, ename, dname
77 FROM emp JOIN dept
78 ON (emp.DEPTNO = dept.DEPTNO)
79 WHERE empno = 7900;
80
81
82
83 REM NON-EQUI JOIN (비등가조인)
84 1. emp table과 salgrade table 간의 관계
85 2. 두개의 테이블사이에 직접 대응하는 열이 없다.
86 3. 이 관계는 =을 제외한 연산자를 사용하여 형성
87
88 SELECT e.ename, e.sal, s.grade
89 FROM emp e, salgrade s
90 WHERE e.sal BETWEEN s.losal AND s.hisal;
91
92
93
94 REM OUTER JOIN
95 1. 포괄조인
96 2. 일치하는 항목이 없는 레코드도 질의할 수 있다.
97
98 SELECT e.ename, e.deptno, d.dname
99 FROM emp e, dept d
100 WHERE e.deptno = d.deptno;
101 --부서 OPERATIONS는 그 부서에서 일하는 사원이 없다. 그래서 표시되지 않는다.
102
103
104 3. 하나이상의 널 행을 생성하여 완전한 테이블의 하나 이상의 행과 조인할 수 있다.
105
106 SELECT e.ename, e.deptno, d.dname
107 FROM emp e RIGHT OUTER JOIN dept d
108 ON e.deptno= d.deptno;
109
110 --LEFT OUTER JOIN을 수행하기 위해 DEPT에 없는 부서 번호 50번 사원을 입력한다.
111 CREATE TABLE emp1
112 AS
113 SELECT * FROM emp;
114
115 INSERT INTO emp(empno, ename, sal, job, deptno)
116 VALUES(8282, 'JACK', 3000, 'ANALYST', 50);
117
118 SELECT e.ename, e.job, e.sal, d.loc, d.dname
119 FROM emp1 e LEFT OUTER JOIN dept d
120 ON (e.deptno = d.deptno)
121
122
123
124 REM SELF JOIN(자체조인)
125 1. 하나의 테이블이 2번 이상 반복적으로 사용되고 참조해야 할 컬럼이 자신의 테이블에 있을 때
126 2. 각 사원의 관리자의 이름을 찾을 때
127 3. 예를 들어, Blake의 관리자 이름을 찾으려면 다음을 수행할 것이다.
128 1)EMP 테이블의 ename 열에서 BLAKE를 찾는다.
129 2)mgr 열에서 BLAKE의 관리자 번호를 찾는다. BLAKE의 관리자 번호는 7839이다.
130 3)ename 열에서 empno 7839인 관리자 이름을 찾는다. King의 사원 번호가 7839이므로 King이 Blake의 관리자이다.
131 4)이 프로세스에서는 테이블을 두 번 검색한다. 첫 번째는 테이블의 ename 열에서 BLAKE와 mgr 값 7839를 찾고 두 번째는 empno
132 열에서 7839를 찾고 ename 열에서 KING을 찾는다.
133
134 SELECT worker.ename, manager.ename
135 FROM emp worker, emp manager
136 WHERE worker.mgr = manager.empno;
137
138 --1.사원 이름 및 사원 번호를 해당 관리자 이름 및 관리자 번호와 함께 표시하고 열 머리글을 각각 "사원이름", "사원번호", "관리자이름",
139 "관리자번호"로 표시하십시오.
140 SELECT employee.ename AS "사원이름",
141 employee.empno AS "사원번호",
142 employer.ename AS "관리자이름",
143 employer.empno AS "관리자번호"
144 FROM emp employee, emp employer
145 WHERE employee.mgr = employer.empno AND employee.deptno = 10;
146

```

147 --2. emp table에서 self join하여 관리자를 출력하되, 아래의 형식에 맞게 출력하시오.
148 --BLAKE 의 관리자는 KING 이다.
149
150
151 --3.사원테이블에서 그들의 관리자보다 먼저 입사한 사원에 대해 이름, 입사일, 관리자 이름, 관리자 입사일을 출력하시오.
152
153
154

REM UNION

156 1. 2개의 쿼리를 위아래로 이어붙여 출력하는 쿼리

```
158 SELECT job, deptno  
159 FROM emp  
160 WHERE sal >= 3000  
161  
162 UNION
```

```
164 SELECT job, deptno  
165 FROM emp  
166 WHERE deptno = 10
```

168 2. 주의할 점

- 1) 위쪽 쿼리와 아래쪽 쿼리 컬럼의 갯수가 동일해야 한다.
- 2) 위쪽 쿼리와 아래쪽 쿼리 컬럼의 데이터타입이 동일해야 한다.
- 3) 결과로 출력되는 컬럼명은 위쪽 쿼리의 컬럼명으로 출력된다.
- 4) ORDER BY절은 제일 아래쪽 쿼리에서만 작성할 수 있다.

REM UNION ALL

178 1. 위아래의 쿼리 결과를 하나의 결과로 출력하는 집합 연산

```
179  
180 SELECT job, deptno  
181 FROM emp  
182 WHERE sal >= 3000  
183  
184 UNION ALL
```

```
186 SELECT job, deptno  
187 FROM emp  
188 WHERE deptno = 10
```

190 2. 주의할 점은 UNION과 동일

191 3. UNION과 UNION ALL의 차이점

- 1) 중복된 데이터를 하나의 고유한 값으로 출력한다.
- 2) 첫 번째 컬럼의 데이터를 기준으로 내림차순으로 정렬하여 출력한다.