Chapter 7. 복수의 테이블 다루기

31강. 집합 연산

1) SQL과 집합

데이터베이스에서의 집합

- 벤 다이어그램 안의 요소 → db에서는 테이블의 행이 요소에 해당
- SELECT 명령 실행 후 반환된 결과 전체는 하나의 집합

2) UNION으로 합집합 구하기

합집합

집합 연산 중, 집합을 서로 더한 것을 가리킴

- 합집합은 두 개의 집합을 더할 때 중복되는 것은 하나로 취급

UNION

- 수학에서 합집합은 A ∪ B로 표현 ⇒ SQL에서 합집합은 'A UNION B'로 표현
- 예제 7-1. UNION으로 합집합 구하기

```
mysql> SELECT * FROM sample71_a;
+------+
| a    |
+------+
|    1 |
|    2 |
|    3 |
+------+
3 rows in set (0.059 sec)

mysql> SELECT * FROM sample71_b;
+------+
| b    |
+------+
|    2 |
|   10 |
|   11 |
+------+
3 rows in set (0.012 sec)
```

- 예제 7-2. 두 개의 SELECT 명령을 UNION해서 합잡합 구하기

```
mysql> SELECT * FROM sample71_a
    -> UNION
    -> SELECT * FROM sample71_b;
+------+
| a    |
+------+
|    1 |
|    2 |
|    3 |
|   10 |
|   11 |
+------+
5 rows in set (0.077 sec)
```

- 한 번의 쿼리 실행으로 두 개의 SELECT 명령이 내부적으로 실행
- SELECT 명령의 실행결과를 합집합으로 계산하여 반환
- 열 구성이 완전히 다른 테이블은 묶을 수 없음
    - 열을 지정하여 집합의 요소가 될 데이터를 맞춘다면 실행 가능
- UNION 시, 나열 순서는 결과에 영향 X

UNION을 사용할 때의 ORDER BY

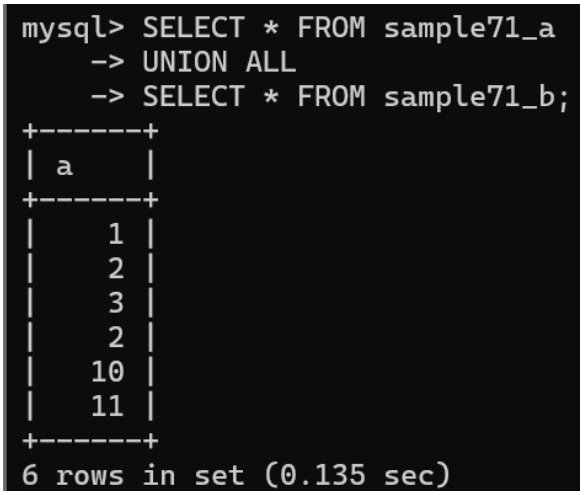
- 각 SELECT명령에 ORDER BY 지정 불가, 마지막 SELECT 명령에만 지정
- 동일하게 별명을 붙여 정렬

```
#UNION에서 OREDER BY 예시
SELECT a AS c FROM sample71_a
UNION
SELECT b AS c FROM sample71_b ORDER BY c;
```

UNION ALL

중복을 제거하지 않고 2개의 SELECT 명령 결과를 합치는 명령

- 예제 7-3. 두 개의 SELECT 명령에 UNION ALL을 적용해 합집합 구하기



- 중복값이 없는 경우에는 UNION보다 좋은 성능을 보임

3) 교집합과 차집합

- 교집합과 차집합은 MySQL에서는 지원X
- 교집합은 **INTERSECT**, 차집합은 **EXCEPT** 사용
- 차집합의 결과가 공집합인지 아닌지에 따라 두 개 집합이 동일한지 파악 가능

32강. 테이블 결합

1) 곱집합과 교차결합

곱집합

두 개의 집합을 곱하는 연산 방법

- '적집합' 또는 '카티전곱(Cartesian product)'이라고도 불림

교차결합(Cross Join)

FROM 구에 복수의 테이블을 지정하면 교차결합을 한다

```
#교차결합
SELECT * FROM 테이블명1, 테이블명 2
```

- 예제 7-5. FROM 구로 곱집합 구하기

```
mysql> SELECT * FROM sample72_x, sample72_y;
+------+------+
| x    | y    |
+------+------+
| C    |    1 |
| B    |    1 |
| A    |    1 |
| C    |    2 |
| B    |    2 |
| A    |    2 |
| C    |    3 |
| B    |    3 |
| A    |    3 |
+------+------+
9 rows in set (0.152 sec)
```

UNION 연결과 결합 연결의 차이

- UNION으로 합집합을 구했을 경우, 세로 방향으로 더해짐
- FROM 구로 테이블을 결합했을 경우, 가로 방향으로 더해짐

2) 내부결합(Inner Join)

교차결합으로 계산된 곱집합에서 원하는 조합을 검색하는 것

- 등결합 이라고도 부름
- 예제 7-9. 상품코드가 같은 행을 검색하기

```
mysql> SELECT * FROM 상품, 재고수
    -> WHERE 상품.상품코드 = 재고수.상품코드;
+----------+--------+----------+------+----------+----------+------------+--------+
| 상품코드 | 상품명 | 메이커명 | 가격 | 상품분류 | 상품코드 | 입고일     | 재고수 |
+----------+--------+----------+------+----------+----------+------------+--------+
| 0001     | 상품1  | 메이커1  |  100 | 식료품   | 0001     | 2014-01-03 |    200 |
| 0002     | 상품2  | 메이커2  |  200 | 식료품   | 0002     | 2014-02-10 |    500 |
| 0003     | 상품3  | 메이커3  | 1980 | 생활용품 | 0003     | 2014-02-14 |     10 |
+----------+--------+----------+------+----------+----------+------------+--------+
3 rows in set (0.224 sec)
```

- 예제 7-10. 검색할 행과 반환할 열 제한하기

```
mysql> SELECT 상품.상품명, 재고수.재고수 FROM 상품, 재고수
    -> WHERE 상품.상품코드 = 재고수.상품코드
    ->  AND 상품.상품분류 = '식료품';
+--------+--------+
| 상품명 | 재고수 |
+--------+--------+
| 상품1  |    200 |
| 상품2  |    500 |
+--------+--------+
2 rows in set (0.020 sec)
```

  - 첫 번째 조건식: **'결합조건'** ; 교차결합으로 계산된 곱집합에서 원하는 조합 검색
  - 두 번째 조건식: 검색 조건

3) INNER JOIN으로 내부결합하기

- 지금까지 설명한 결합 방법 정리
  1. FROM 구에서 테이블을 복수로 지정해 가로 방향으로 테이블 결합
  2. 교차결합을 하면 곱집합으로 계산
  3. WHERE 조건을 지정해 곱집합에서 필요한 조합만 검색 가능
- 최근에는 **INNER JOIN**키워드를 사용한 결합방법으로 통용됨

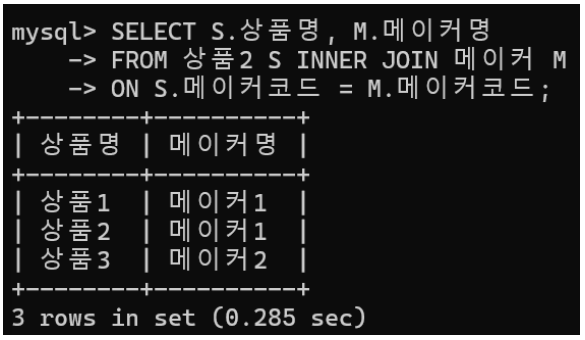
```
#위의 예제를 INNER JOIN으로 바꿔보기
SELECT 상품.상품명, 재고수.재고수
   FROM 상품 INNER JOIN 재고수
      ON 상품.상품코드 = 재고수.상품코드
   WHERE 상품.상품분류 = '식료품';
```

```
#내부결합
```

```
SELECT * FROM 테이블명1 INNER JOIN 테이블명2 ON 결합조건
```

4) 내부결합을 활용한 데이터 관리

- 예제 7-12. 상품 테이블과 메이커 테이블을 내부결합하기



```
mysql> SELECT S.상품명, M.메이커명
    -> FROM 상품2 S INNER JOIN 메이커 M
    -> ON S.메이커코드 = M.메이커코드;
+--------+----------+
| 상품명 | 메이커명 |
+--------+----------+
| 상품1  | 메이커1  |
| 상품2  | 메이커1  |
| 상품3  | 메이커2  |
+--------+----------+
3 rows in set (0.285 sec)
```

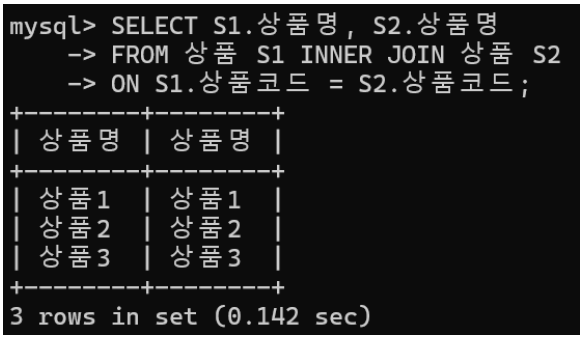
외부키

다른 테이블의 기본키를 참조하는 열

자기결합(Self Join)

테이블에 별명을 붙일 수 있는 기능을 이용해 같은 테이블끼리 결합하는 것

- 예제 7-13. 상품 테이블을 자기결합하기



```
mysql> SELECT S1.상품명, S2.상품명
    -> FROM 상품 S1 INNER JOIN 상품 S2
    -> ON S1.상품코드 = S2.상품코드;
+--------+--------+
| 상품명 | 상품명 |
+--------+--------+
| 상품1  | 상품1  |
| 상품2  | 상품2  |
| 상품3  | 상품3  |
+--------+--------+
3 rows in set (0.142 sec)
```

- 자기결합을 사용하는 일은 드묾
- 자기 자신의 기본키를 참조하는 열을 자기 자신이 가지는 데이터 구조로 되어 있는 경우에 자주 사용

5) 외부결합

'어느 한 쪽에만 존재하는 데이터행을 어떻게 다룰지'를 변경할 수 있는 결합 방법

- 교차결합으로 결합 조건을 지정하여 검색한다는 기본적인 방식은 내부결합과 동일
- 예제 7-14. 내부결합에서는 상품코드가 009인 상품이 제외된다

```
mysql> SELECT 상품3.상품명, 재고수.재고수
    -> FROM 상품3 INNER JOIN 재고수
    -> ON 상품3.상품코드=재고수.상품코드
    -> WHERE 상품3.상품분류='식료품';
+--------+--------+
| 상품명 | 재고수 |
+--------+--------+
| 상품1  |    200 |
| 상품2  |    500 |
+--------+--------+
2 rows in set (0.206 sec)
```

- 외부결합 시, 결합하는 테이블 중에 어느쪽을 기준으로 할지 결정 가능
- 결합 기준을 왼쪽으로 할 시, **LEFT JOIN**사용
- 예제 7-15. 외부결합으로 상품코드 009인 상품도 결과에 포함하기

```
mysql> SELECT 상품3.상품명, 재고수.재고수
    -> FROM 상품3 LEFT JOIN 재고수
    -> ON 상품3.상품코드 = 재고수.상품코드
    -> WHERE 상품3.상품분류='식료품';
+----------+--------+
| 상품명   | 재고수 |
+----------+--------+
| 상품1    |    200 |
| 상품2    |    500 |
| 추가상품 |   NULL |
+----------+--------+
3 rows in set (0.138 sec)
```

  - 재고수 테이블에는 009에 대한 데이터가 없으므로 값이 NULL로 표시됨
- 결합 기준을 오른쪽으로 지정할 시, **RIGHT JOIN 사용**

구식방법에서의 외부결합과 표준 SQL ⇒ 구식 결합방법은 사용X

- 구식 결합방법(Oracle 예)
  - FROM 구에 결합 조건 기술X, WHERE 구로 결합 조건 지정
  - 외부 결합 시, 특별한 연산자(+) 사용

- 예제 7-16. Oracle에서 구식 외부 결합으로 009의 상품을 결과에 포함하기

```
SELECT 상품3.상품명, 재고수.재고수
  FROM 상품3, 재고수
  WHERE 상품3.상품코드=재고수.상품코드(+)
    AND 상품3.상품분류='식료품';
```

- SQL Server에서는 특수한 연산자(*= 또는 =*)를 사용하여 외부 결합
- 현재는 표준화로, 내부 결합은 INNER JOIN, 외부 결합은 RIGHT JOIN/LEFT JOIN 사용

33강. 관계형 모델

1) 관계형 모델

- **릴레이션(Relation)**

  관계형 모델의 릴레이션은 **테이블**에 해당

- **속성(attribute)**

  SQL에서의 속성은 열에 해당

  - 속성은 속성 이름과 형 이름으로 구성

- **튜플(tuple)**

  SQL에서의 행은 관계형 모델에서 튜플

- **관계대수**

  릴레이션에 대한 연산이 집합의 대한 연산에 대응된다는 이론

  - 기본 규칙
    - 하나 이상의 관계를 바탕으로 연산한다.
    - 연산한 결과, 반환되는 것 또한 관계이다.
    - 연산을 중첩 구조로 실행해도 상관없다.

2) 관계형 모델과 SQL

(자주 사용되는 릴레이션 연산 방법을 SQL 명령과 비교하며 설명)

연산 방법	설명	SQL 명령
합집합(union)	릴레이션끼리의 덧셈	UNION
차집합(difference)	릴레이션끼리의 뺄셈	EXCEPT
교집합(intersection)	릴레이션끼리의 공통부분	INTERSECT
곱집합(cartesian product)	릴레이션끼리의 대진표를 조합하는 연산	-FROM구에 복수의 테이블 -CROSS JOIN
선택(selection)	튜플의 추출	WHERE구에 조건 지정
투영(projection)	속성의 추출	SELECT구에 결과를 반환할 열 지정
결합(join)	릴레이션끼리 교차결합해 계산된 곱집합에서 결합조건을 만족하는 튜플을 추출하는 연산	내부결합