

Chapter 15. 집합 연산자

이번 장에서는 집합 연산자에 대하여 알아보고 집합 연산자를 이용하여 여러 개의 쿼리 문장을 하나의 쿼리 문장으로 결합하는 방법을 설명한다.

집합 연산자의 종류

집합 연산자는 두 개의 쿼리 문장을 결합하며 다음과 같은 연산을 할 수 있다.

표 15-1. 집합 연산자

연산자	결과 값
UNION	각 쿼리에 의해서 선택된 결과 중에 중복 행을 제거하고 출력
UNION ALL	각 쿼리에 의해서 선택된 모든 결과를 출력, 중복 허용
INTERSECT	두 개의 쿼리를 모두 만족하는 결과 중에 중복 행을 제거하고 출력
MINUS	첫 번째 쿼리에 의해서 선택된 결과에서 두 번째 쿼리에 의해서 선택된 결과를 제거하고 출력

집합 연산자의 사용방법을 학습하기 위해 다음과 같은 임시 테이블을 작성하자.

```
CREATE TABLE A(VAL CHAR);
INSERT INTO A VALUES ('A');
INSERT INTO A VALUES ('B');
INSERT INTO A VALUES ('C');
INSERT INTO A VALUES ('D');
INSERT INTO A VALUES ('E');

CREATE TABLE B(NUM NUMBER, VAL CHAR);
INSERT INTO B VALUES (1, 'C');
INSERT INTO B VALUES (2, 'D');
INSERT INTO B VALUES (3, 'E');
INSERT INTO B VALUES (4, 'F');
INSERT INTO B VALUES (5, 'G');
```

UNION

UNION 연산자는 각각의 쿼리에 의해서 선택된 모든 행들에 대하여 중복을 제거하고 출력한다. UNION 연산자를 사용할 때 다음과 같은 사항을 주의해야 한다.

- 각각의 SELECT 문장에 의해서 선택된 컬럼의 개수와 데이터 타입은 반드시 일치해야 하지만, 컬럼명은 같을 필요가 없다.
- UNION 연산자는 선택된 모든 컬럼들에 대하여 적용된다.
- 각각의 쿼리 결과에서 중복을 제거하는 과정에 NULL 값은 무시된다.
- IN 연산자가 UNION 연산자보다 우선순위가 높다.
- 기본적으로 SELECT 문장의 첫 번째 컬럼 기준으로 오름차순 정렬된다.

다음은 A 테이블과 B 테이블을 UNION 연산한 결과이다.

```
SQL> SELECT VAL FROM A
2 UNION
3 SELECT VAL FROM B;
```

```
V
-
A
B
C
D
E
F
G
```

7 개의 행이 선택되었습니다.

UNION ALL

UNION ALL 연산자는 각각의 쿼리에 의해서 선택된 모든 행들을 출력한다. UNION ALL 연산자를 사용 할 때 다음과 같은 사항을 주의해야 한다.

- UNION과는 달리 중복된 행을 제거하지 않으며, 디폴트로 출력결과를 정렬하지 않는다.
- DISTINCT 키워드는 사용 할 필요가 없다.

다음은 A 테이블과 B 테이블을 UNION ALL 연산한 결과이다.

```
SQL> SELECT VAL FROM A
2 UNION ALL
3 SELECT VAL FROM B;
```

```
V
-
A
B
C
D
E
C
D
E
F
G
```

10 개의 행이 선택되었습니다.

INTERSECT

INTERSECT 연산자는 각각의 쿼리에 공통적으로 포함되어 있는 행들을 출력한다. INTERSECT 연산자를 사용 할 때 다음과 같은 사항을 주의해야 한다.

- 각각의 SELECT 문장에 의해서 선택된 컬럼의 개수와 데이터 타입은 반드시 일치해야 하지만, 컬럼명은 같을 필요가 없다.
- 각각의 쿼리 작성 순서는 연산 결과를 변경하지 않는다.
- INTERSECT 연산자는 NULL 값을 무시하지 않는다.

다음은 A 테이블과 B 테이블을 INTERSECT 연산한 결과이다.

```
SQL> SELECT VAL FROM A
2  INTERSECT
3  SELECT VAL FROM B;

V
-
C
D
E
```

MINUS

MINUS 연산자는 첫 번째 쿼리 결과에서 두 번째 쿼리 결과에 포함된 행들을 제거하고 출력한다. MINUS 연산자를 사용 할 때 다음과 같은 사항을 주의해야 한다.

- 각각의 SELECT 문장에 의해서 선택된 컬럼의 개수와 데이터 타입은 반드시 일치해야 하지만, 컬럼명은 같을 필요가 없다.

다음은 A 테이블과 B 테이블을 MINUS 연산한 결과이다.

```
SQL> SELECT VAL FROM A
2  MINUS
3  SELECT VAL FROM B;

V
-
A
B
```

집합 연산자 사용 지침

집합 연산자를 사용할 때는 각각의 SELECT 문장의 컬럼의 개수와 데이터 타입이 정확히 일치해야 하며, 집합 연산자의 처리 순서를 변경하려면 괄호를 이용하면 된다. 집합 연산자가 사용된 쿼리 문장에서 ORDER BY 절을 사용할 때는 문장의 제일 마지막에 기술하며, 첫 번째 쿼리 문장의 컬럼명, 별칭, 또는 컬럼 순번을 지정 할 수 있다.

Oracle 서버에서 집합 연산자는 다음과 같은 특징을 갖는다.

- Oracle에서 집합 연산자는 UNION ALL 연산을 제외한 모든 연산에서 중복된 행을 자동

으로 제거한다.

- 첫 번째 쿼리의 컬럼명이 결과에 표시된다.
- UNION ALL 연산을 제외하고 모든 출력은 기본적으로 오름차순 정렬된다.

집합 연산자를 사용 할 때, 각각의 쿼리 문장에서 컬럼의 개수와 데이터 타입이 불일치 되는 경우 컬럼을 추가하거나 데이터 타입 변환 함수를 사용하여 일치시키면 된다.

```
SQL> SELECT VAL, NULL FROM A
2  UNION
3  SELECT VAL, NUM FROM B;
```

```
V      NULL
-----
```

```
A
B
C      1
C
D      2
D
E      3
E
F      4
G      5
```

10 개의 행이 선택되었습니다.

집합 연산자는 기본적으로 첫 번째 쿼리의 첫 번째 컬럼을 기준으로 오름차순 정렬되지만, 각 쿼리 결과별로 정렬하려면 별도의 컬럼을 추가하고 ORDER BY에 기술해주면 된다.

```
SQL> COL C NOPRINT
SQL> SELECT VAL, 1 C FROM A
2  UNION
3  SELECT VAL, 2 FROM B
4  ORDER BY 2;
```

```
V
-
A
B
C
D
E
C
D
E
F
G
```

10 개의 행이 선택되었습니다.

복습

1. 다음 연산자 중 출력결과가 정렬되지 않는 집합 연산자는?
 - a. UNION
 - b. UNION ALL
 - c. INTERSECT
 - d. MINUS

2. 다음 연산자 중 쿼리의 작성 순서가 결과에 영향을 미치는 집합 연산자는?
 - a. UNION
 - b. UNION ALL
 - c. INTERSECT
 - d. MINUS

3. 집합 연산자를 사용 할 때 주의해야 할 사항이 아닌 것은?
 - a. 각각의 쿼리에서 컬럼의 개수는 일치해야 한다.
 - b. 각각의 쿼리에서 컬럼의 데이터 타입은 일치해야 한다.
 - c. 각각의 쿼리에서 컬럼명은 일치해야 한다.

