

## Chapter 19. 계층형 쿼리

이번 장에서는 계층형 쿼리에 대한 개념과 계층형 쿼리를 이용하여 트리 구조의 보고서를 작성하는 방법을 알아본다. 또한, 트리 구조에서 특정 노드(Node) 또는 브랜치(Branch)를 제거하고 검색하는 방법을 설명한다.

### 트리 구조

트리 구조를 설명하기에 앞서 사원 테이블의 데이터를 확인해보자.

```
SQL> SELECT * FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

14 개의 행이 선택되었습니다.

앞선 내용에서 사원 테이블의 MGR 컬럼은 관리자사번이라고 설명하였다. 이 관리자사번은 사원들의 상하 수직관계를 데이터베이스에 저장하기 위해 추가한 컬럼인데 의미하는 바는 다음과 같다. 즉, 사번이 7369인 SMITH의 관리자는 사번이 7902번이며, 7902번 FORD의 관리자는 사번이 7566번이고, 7566번 JONES의 관리자 사번이 7839인 KING을 저장한 것이다. 이러한 관계를 우리가 평소에 많이 볼 수 있는 트리 구조로 표현하면 다음과 같다.

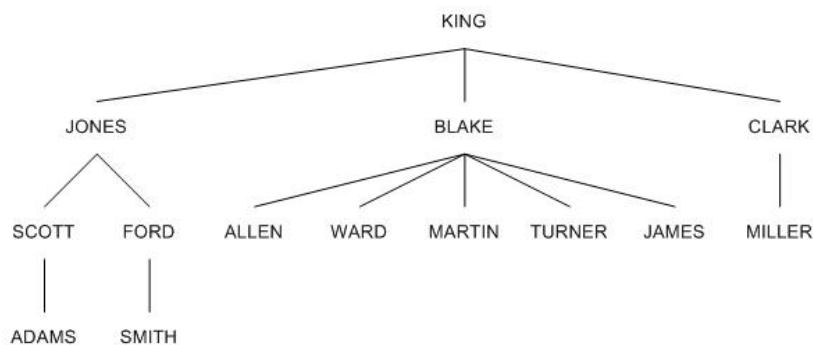


그림 19-1. 트리 구조

## 계층형 쿼리의 작성 방법

위 그림과 같은 트리 구조의 데이터를 검색하는 방법이 계층형 쿼리이며, 문법은 다음과 같다.

```
SELECT [LEVEL], column, expr ...
FROM table
[WHERE condition(s)]
[START WITH condition(s)]
[CONNECT BY PRIOR condition(s)];
```

여기서, LEVEL은 계층형 쿼리에서 출력되는 가상 컬럼으로 1은 루트, 2는 루트의 자식, 3은 루트의 자식의 자식을 표시한다. START WITH는 트리 구조의 루트에 위치할 행을 지정하는 것으로 계층형 질의가 시작되는 행을 의미한다. CONNECT BY에는 트리 구조에서 부모 행과 자식 행 사이의 관계를 조건으로 기술하는데 PRIOR 뒤에 기술된 컬럼은 계층형 질의를 진행하면서 바로 전에 읽은 행을 의미한다.

먼저, 계층형 쿼리를 작성하려면 트리 구조의 최상단, 즉 루트에 위치할 행을 검색 할 수 있는 조건을 START WITH에 기술한다.

```
START WITH column1 = value
```

예를 들어, 사원 테이블에서 루트에 위치할 사원의 이름이 'KING'이라면 다음과 같이 기술한다.

```
START WITH ENAME = 'KING'
```

다음은 계층형 쿼리의 진행 방향을 기술하는데 트리 구조에서 START WITH로 지정한 행부터 하향식(Top-down)으로 전개하려면 부모 컬럼을 *column1*, 자식 컬럼을 *column2*를 기술하고, 상향식(Bottom-up)으로 전개하려면 자식 컬럼을 *column1*, 부모 컬럼을 *column2*에 기술한다.

```
CONNECT BY PRIOR column1 = column2
```

예를 들어, 사원 테이블에서 트리 구조의 루트에 위치한 'KING'으로부터 하향식 전개하려면 다음과 같이 기술해야 한다.

```
CONNECT BY PRIOR EMPNO = MGR
```

사원 테이블에서 사원명이 'KING'인 직원부터 하향식 전개하는 방법은 다음과 같다.

```
SQL> SELECT EMPNO, ENAME, MGR
2  FROM EMP
3  START WITH ENAME = 'KING'
4  CONNECT BY PRIOR EMPNO = MGR;
```

EMPNO	ENAME	MGR
7839	KING	
7566	JONES	7839
7788	SCOTT	7566
7876	ADAMS	7788
7902	FORD	7566
7369	SMITH	7902
7698	BLAKE	7839
7499	ALLEN	7698
7521	WARD	7698
7654	MARTIN	7698
7844	TURNER	7698
7900	JAMES	7698
7782	CLARK	7839
7934	MILLER	7782

14 개의 행이 선택되었습니다.

반대로 사원 테이블에서 사원명이 'SMITH'인 직원부터 상향식 전개하는 방법은 다음과 같다.

```
SQL> SELECT EMPNO, ENAME, MGR
2  FROM EMP
3  START WITH ENAME = 'SMITH'
4  CONNECT BY PRIOR MGR = EMPNO;
```

EMPNO	ENAME	MGR
7369	SMITH	7902
7902	FORD	7566
7566	JONES	7839
7839	KING	

## LEVEL

계층형 쿼리에서 사용 할 수 있는 가상 컬럼 LEVEL은 트리 구조에서 계층을 나타낸다. 즉, LEVEL 1은 루트, 2는 루트의 자식, 3은 루트의 자식의 자식을 의미한다. 예를 들어, 앞 그림에서 'KING'은 LEVEL=1, 'JOHN'은 LEVEL=2, 'FORD'는 LEVEL=3, 'SMITH'는 LEVEL=4이다.

계층형 쿼리에서 가상 컬럼인 LEVEL과 LPAD 함수를 사용하면 좀 더 보기 쉬운 보고서를 작성 할 수 있다. 즉, LEVEL 값에 따라 LPAD 함수를 이용하여 출력될 행의 앞에 들여쓰기를 하면 계층형 쿼리 결과를 알아보기 쉬운 형태로 만들 수 있다.

```
SQL> SELECT LPAD(ENAME, LENGTH(ENAME)+(LEVEL*2)-2, ' ') NAME
2 FROM EMP
3 START WITH ENAME = 'KING'
4 CONNECT BY PRIOR EMPNO = MGR;
```

NAME

```
-----
KING
  JONES
    SCOTT
      ADAMS
        FORD
          SMITH
    BLAKE
      ALLEN
      WARD
      MARTIN
      TURNER
      JAMES
    CLARK
      MILLER
```

14 개의 행이 선택되었습니다.

## 트리 구조의 노드(Node) 및 브랜치(Branch) 제거

'KING'으로부터 하향식 전개된 계층형 쿼리 결과에서 'FORD'를 제외하고 출력하려면 WHERE 절에서 제거할 노드를 제외 할 수 있는 조건을 기술해주면 된다.

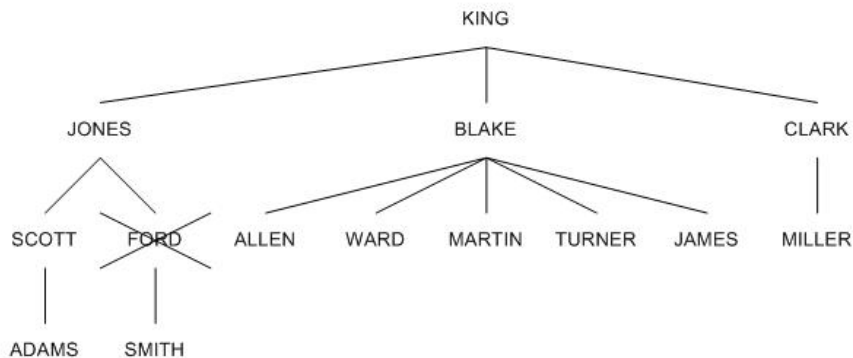


그림 19-2. 노드 제거

```

SQL> SELECT LPAD(ENAME, LENGTH(ENAME)+(LEVEL*2)-2, ' ') NAME
2  FROM EMP
3  WHERE ENAME != 'FORD'
4  START WITH ENAME = 'KING'
5  CONNECT BY PRIOR EMPNO = MGR;

```

NAME

```

KING
  JONES
    SCOTT
      ADAMS
      SMITH
  BLAKE
    ALLEN
    WARD
    MARTIN
    TURNER
    JAMES
  CLARK
    MILLER

```

13 개의 행이 선택되었습니다.

또한, 아래 그림과 같이 트리 구조에서 'FORD'의 브랜치를 제거하고 계층형 쿼리를 진행하고자 하는 경우에는 CONNECT BY 절 뒤에 제거할 브랜치의 최고 관리자를 제외 할 수 있는 조건을 기술하면 된다.

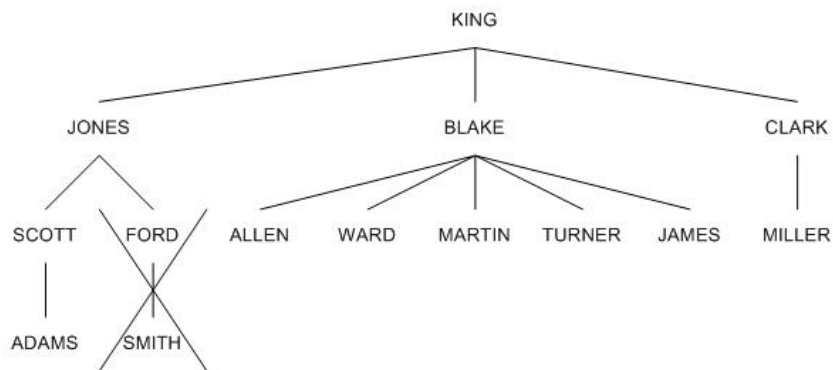


그림 19-3. 브랜치 제거

```
SQL> SELECT LPAD(ENAME, LENGTH(ENAME)+(LEVEL*2)-2, ' ') NAME
2  FROM EMP
3  START WITH ENAME = 'KING'
4  CONNECT BY PRIOR EMPNO = MGR
5  AND ENAME != 'FORD';
```

NAME

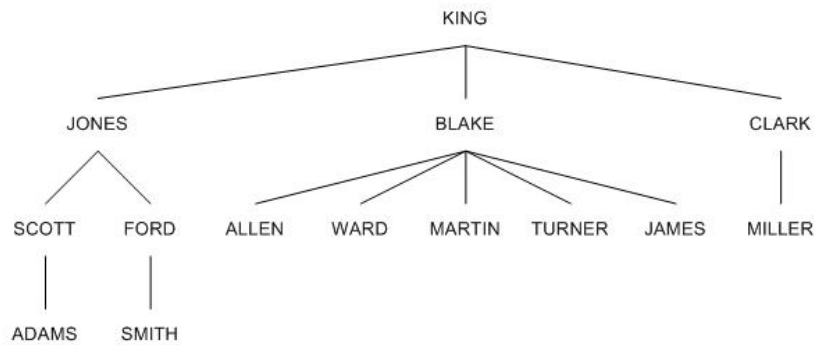
-----

```
KING
  JONES
    SCOTT
      ADAMS
    BLAKE
      ALLEN
        WARD
          MARTIN
            TURNER
              JAMES
                CLARK
                  MILLER
```

12 개의 행이 선택되었습니다.

## 복습

사원 테이블과 아래 트리 구조를 참고하여 적절한 쿼리 문장을 작성하시오.



1. 계층형 쿼리를 이용하여 'JONES'의 부하직원을 모두 출력하시오.
2. 계층형 쿼리를 이용하여 'ADAMS'의 상관을 모두 출력하시오.
3. 계층형 쿼리를 이용하여 'KING'의 부하직원을 출력하시오. 단, 'BLAKE'와 'BLAKE'의 부하직원들은 제외 할 것.

