

Hierarchikus adatszerkezet megjelenítése

Az adattáblák logikai szerkezetében gyakran fordul elő hierarchikus (faszerkezetű) kapcsolat az adatok között (például főnök–beosztott, családfa stb.). Ilyen esetekben az egyik tipikus lekérdezési feladat e hierarchikus kapcsolat megjelenítése. Az alábbiakban két módszert is bemutatunk.

Az egyik (a korrelált lekérdezés) során az egyetlen adattáblára két különböző néven hivatkozunk, és így a hierarchikus kapcsolatot e „két” tábla között hagyományos többtáblás lekérdezésként térképezzük fel.

A másik módszer az SQL nyelv kifejezetten a hierarchikus adatszerkezetek szemléletes megjelenítésére szolgáló nyelvi eszközének, a SELECT utasításba épített CONNECT BY utasításrésznek a használatán alapul.

KORRELÁLT LEKÉRDEZÉS

3.2. példa

Listázzuk ki az emp tábla alapján a főnökökhöz tartozó beosztottak nevét és részlegük azonosítóját célszerű módon.

1. megoldás

A feladat megoldható az emp tábla önmagára való hivatkozásával, vagyis az alábbi, logikailag többtáblás (korrelált) lekérdezéssel:

```
SELECT főnök.ename      AS "Főnök neve",
       dolgozó.ename   AS "Dolgozó neve",
       dolgozó.deptno AS "Részlegazonosító"
FROM emp dolgozó,
     emp főnök
WHERE főnök.empno = dolgozó.mgr
ORDER BY főnök.ename,
         dolgozó.ename;
```

Főnök neve	Dolgozó ne	Részlegazonosító
BLAKE	ALLEN	30
BLAKE	JAMES	30
BLAKE	MARTIN	30
BLAKE	TURNER	30
BLAKE	WARD	30
CLARK	MILLER	10
FORD	SMITH	20
JONES	FORD	20
JONES	SCOTT	20
KING	BLAKE	30
KING	CLARK	10

KING	JONES	20
SCOTT	ADAMS	20

13 sor kijelölve.

A fenti listában csak 13 sor szerepel, mivel King-nek nincs főnöke.

HIERARCHIALEKÉRDEZÉS (CONNECT BY)

Az előző pontban elkezdett feladat megoldása előtt bemutatjuk a CONNECT BY utasításrész használatát (pontos ismertetést lásd [12]):

```
SELECT SzelekciósLista
FROM tábla
[WHERE LogikaiFeltétel]
CONNECT BY { PRIOR kifejezés = kifejezés |
            kifejezés = PRIOR kifejezés }
[START WITH kifejezés = kifejezés]
[ORDER BY ...];
```

ahol a korábban nem ismertetett kulcsszavak jelentése:

- a *SzelekciósLista* tartalmazhatja a LEVEL pseudooszlopot (belső változót), amely megadja, hogy az aktuálisan listázott elem a hierarchia mely szintjén található,
- a START WITH utasításrészben adjuk meg, hogy a hierarchikus szerkezet feldolgozása hol kezdődjön,
- a WHERE utasításrészben megadott logikai feltétel csak konkrét sorok kiszűrésére alkalmas, leszármazottai és ősei (szülőcsúctól függően) kiszűrésére nem,
- a CONNECT BY utasításrészben megadott feltételt használjuk a sorok és leszármazottainak kiszűrésére,
- a PRIOR meghatározza a bejárás szerinti szülőcsúcsot (függetlenül attól, hogy az egyenlőségjel jobb vagy bal oldalán áll).

A 3.2. példa megoldásának folytatása

2. megoldás

Tekintsük ezután az előző feladat megoldását a CONNECT BY utasításrész használatával:

```
SELECT LEVEL AS szint,
       ename AS dolgozó,
       deptno AS részleg
FROM emp
CONNECT BY mgr = PRIOR empno
START WITH UPPER(ename) = 'KING'
ORDER BY deptno;
```

SZINT	DOLGOZÓ	RÉSZLEG
1	KING	10
2	CLARK	10
3	MILLER	10
2	JONES	20
3	SCOTT	20
4	ADAMS	20
3	FORD	20
4	SMITH	20
2	BLAKE	30
3	ALLEN	30
3	WARD	30
3	MARTIN	30
3	TURNER	30
3	JAMES	30

14 sor kijelölve.

A fenti listában most már 14 sor szerepel, mivel a King is a hierarchikus szervezet tagja.

3. megoldás

Ha az előző példabeli listához hasonló eredményt szeretnénk látni, akkor a

```
SELECT LEVEL AS szint,
       ename AS dolgozó,
       deptno AS részleg
FROM emp
WHERE mgr IS NOT NULL
CONNECT BY mgr = PRIOR empno
START WITH UPPER(ename) = 'KING'
ORDER BY deptno;
```

utasítást használva kapjuk:

SZINT	DOLGOZÓ	RÉSZLEG
2	CLARK	10
3	MILLER	10
2	JONES	20
3	SCOTT	20
4	ADAMS	20
3	FORD	20
4	SMITH	20

2	BLAKE	30
3	ALLEN	30
3	WARD	30
3	MARTIN	30
3	TURNER	30
3	JAMES	30

13 sor kijelölve.

4. megoldás

Ha a hierarchikus szerkezet egy ágát az ellenkező irányból kívánjuk bejárni, például Smith-től kezdődően, akkor a

```
SELECT LEVEL AS szint,
       ename AS dolgozó,
       deptno AS részleg
FROM emp
CONNECT BY PRIOR mgr = empno
START WITH UPPER(ename) = 'SMITH'
ORDER BY deptno;
```

vagy

```
SELECT LEVEL AS szint,
       ename AS dolgozó,
       deptno AS részleg
FROM emp
CONNECT BY empno = PRIOR mgr
START WITH UPPER(ename) = 'SMITH'
ORDER BY deptno;
```

utasítással kapott lista:

SZINT	DOLGOZÓ	RÉSZLEG
4	KING	10
1	SMITH	20
2	FORD	20
3	JONES	20