

# Rekurzió illetve hierarchikus lekérdezések

## ■ Rekurzió:

### □ hierarchikus lekérdezés:

- **CONNECT BY** használata SELECT-ben

### □ rekurzív logikai adatmodell (DATALOG):

- **WITH** utasítás

- rekurzív függvények, eljárások

## ■ Hierarchikus lekérdezések

- a rekordok között megadható egy **logikai hierarchia** (aciklikus gráf),

- egy rekurzív feltétel teljesülése határozza meg, hogy melyik szülőrekordnak melyek a gyerekrekordjai

- megadható egy kezdőfeltétel, amely meghatározza, hogy mely rekordok helyezkednek el a legfelső szinten.

- select ... **start with** kezdő feltétel

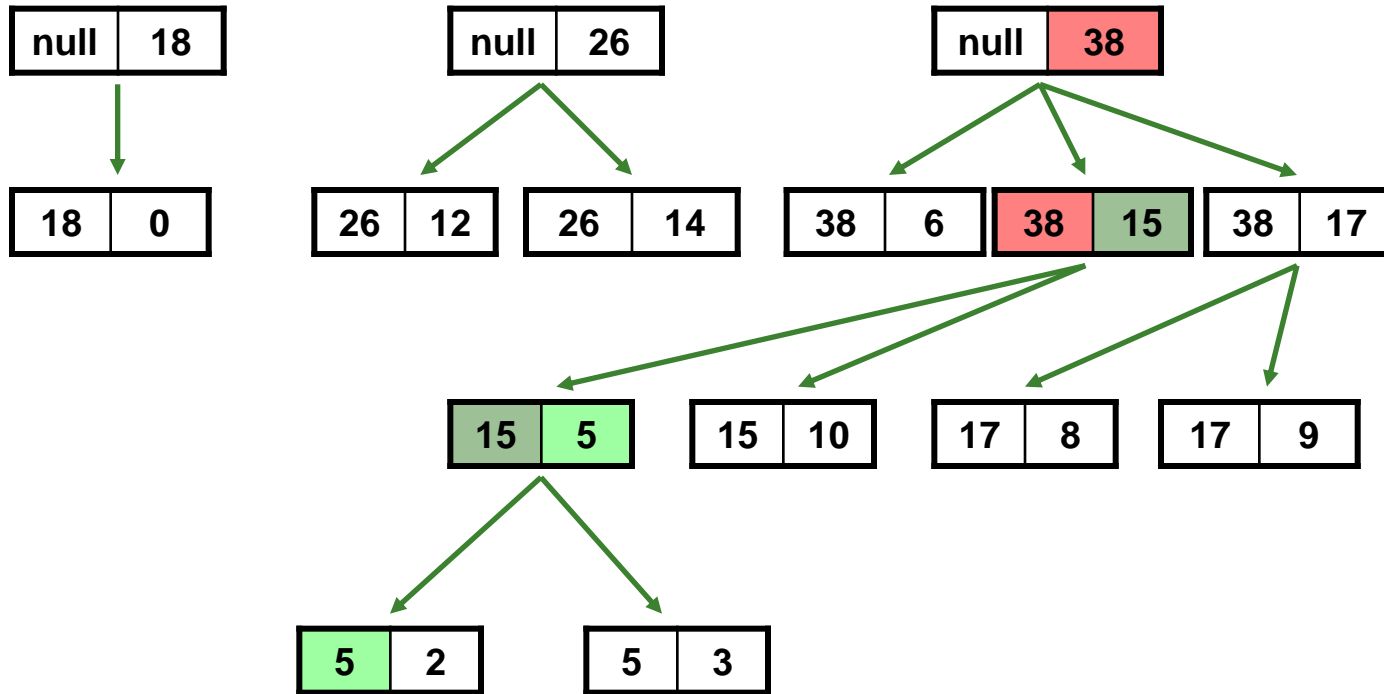
**connect by prior** rekurzív feltétel;

# Hierarchikus lekérdezések

create table test\_connect\_by

(parent number, child number, **constraint** uq\_tcb unique (child) );

/\* minden gyereknek egyértelmű, hogy ki a szülője \*/



parent	child
5	2
5	3
18	0
17	8
17	9
26	12
26	14
15	5
15	10
38	6
38	15
38	17
null	18
null	26
null	38

- 4 szinten helyezkednek el a rekordok.
- A rekordhoz tartozó szintek számát egy pseudooszlop, a **level** tartalmazza.
- A gyökererekordok esetén a **level=1**.

# Hierarchikus lekérdezések

A hierarchia szerinti bejárást a CONNECT BY PRIOR használatával lehet megadni:

```
select lpad(' ',2*(level-1)) || to_char(child) s
      from test_connect_by
start with parent is null
connect by prior child = parent
order siblings by child;
```

- Mélységi bejárás
- Szinteken belüli sorrendet az ORDER SIBLINGS BY határozza meg.
- A szülő és gyerek rekordok közötti kapcsolat nemcsak oszlopegyezésen alapulhat.
- A PRIOR oszlopnév a szülőrekord oszlopát jelenti.

18

0

26

12

14

38

6

15

5

2

3

10

17

8

9

# Hierarchikus lekérdezések

A következő pszeudokód felel meg a kiértékelésnek:

```
for rec in (select * from some_table order by syblings_order) loop
  if FULLFILLS_START_WITH_CONDITION(rec) then
    RECURSE(rec, rec.child);
  end if;
end loop;
```

```
procedure RECURSE (rec in MATCHES_SELECT_STMT, new_parent IN field_type)
is
begin
  APPEND_RESULT_LIST(rec);
  for rec_recurse in (select * from some_table order by syblings_order) loop
    if FULLFILLS_CONNECT_BY_CONDITION(rec_recurse.child,new_parent)
    then RECURSE(rec_recurse,rec_recurse.child);
    end if;
  end loop;
end procedure RECURSE;
```