

Rekurzió az SQL-ben

Tankönyv: Ullman-Widom:
Adatbázisrendszerek Alapvetés
Második, átdolgozott kiadás,
Panem, 2009

10.2. Rekurzió az SQL-ben,
az „Eljut”-feladat
Oracle-ben és PL/SQL-ben



Az „Eljut feladat” SQL-99 szabványban

Tankönyv 10.2. fejezet példája (az ELJUT feladat)

➤ Jaratok(legitarsasag, honnan, hova, koltseg, indulas, erkezes) táblában repülőjáratok adatait tároljuk. Mely városokba tudunk eljutni Bp-ről?

➤ **WITH RECURSIVE** Eljut AS

(SELECT honnan, hova FROM Jaratok

UNION

SELECT Eljut.honnan, Jaratok.hova

FROM Eljut, Jaratok

WHERE Eljut.hova = Jaratok.honnan)

SELECT hova FROM Eljut WHERE honnan='Bp';

Oracle megoldások: WITH utasítással

- Az **Oracle SQL** a WITH RECURSIVE utasítást nem támogatja, ott másképpen oldották meg WITH utasítással (Oracle 11gR2 verziótól)
- **with** eljut (honnan, hova) as
(select honnan, hova from jaratok
union all
select jaratok.honnan, eljut.hova
from jaratok, eljut
where jaratok.hova=eljut.honnan
)
SEARCH DEPTH FIRST BY honnan SET SORTING
CYCLE honnan SET is_cycle TO 1 DEFAULT 0
select distinct honnan, hova from eljut order by honnan;

Oracle megoldások: connect by

- SELECT DISTINCT hova FROM jaratok
WHERE HOVA <> 'DAL'
START WITH honnan = 'DAL'
CONNECT BY **NOCYCLE** PRIOR hova = honnan;
- SELECT LPAD(' ', 4*level) || honnan, hova,
level-1 Atszallasok,
sys_connect_by_path(honnan||'-'>||hova, '/'),
connect_by_isleaf, connect_by_iscycle
FROM jaratok
START WITH honnan = 'SF'
CONNECT BY **NOCYCLE** PRIOR hova = honnan;

Rekurzív Eljut feladat PSM-ben ---1

- Az ELJUT feladatot a gyakorlaton oldjuk meg Oracle PL/SQL-ben, itt **csak a vázlata PSM-ben**
- A ciklus során ellenőrizni kell, hogy addig hajtsuk végre a ciklust, amíg növekszik az eredmény (Számláló)
- **DECLARE** RegiSzamlalo Integer;
UjSzamlalo Integer;
- Deklarációs rész után **BEGIN ... END;** között az utasítások, először az eljut táblának kezdeti értéket adunk (a megvalósításnál az INSERT-nél figyelni, hogy ne legyenek ismétlődő sorok: `select distinct delete from eljut; insert into eljut (SELECT distinct honnan, hova FROM jaratok);`)

Rekurzív Eljut feladat PSM-ben ---2

- Szamlalo változóknak adunk kiindulási értéket:

```
SET RegiSzamlalo = 0;
```

```
select count(*) into UjSzamlalo from eljut;
```

- A ciklust addig kell végrehajtani, amíg növekszik az eredmény (Szamlalo) duplikátumokra figyelni!

LOOP

```
insert into eljut (lásd a köv.oldalon...)
```

```
select count(*) into UjSzamlalo from eljut;
```

```
EXIT WHEN UjSzamlalo = RegiSzamlalo;
```

```
SET RegiSzamlalo = UjSzamlalo;
```

```
END LOOP;
```

Rekurzív Eljut feladat PSM-ben ---3

- Az eljut tábla növelése a ciklusban, figyelni kell a duplikátumokra, csak olyan várospárokat vegyünk az eredményhez, ami még nem volt!

insert into eljut

```
(select distinct eljut.honnan, jaratok.hova  
from eljut, jaratok --- *from (lineáris rekurzió)  
where eljut.hova = jaratok.honnan  
and (eljut.honnan,jaratok.hova)  
NOT IN (select * from eljut));
```

- Megjegyzés: PSM-ben a **nem-lineáris rekurzió** is megengedett: **from eljut e1, eljut e2 ---*from-ban**