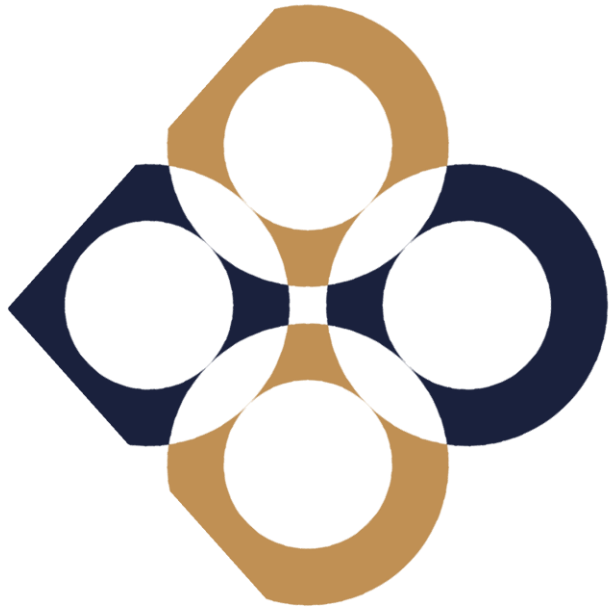
BUDAPESTI CORVINUS EGYETEM

Adatbázisok

Gyakorlat 04 –
A csoportosítás speciális lehetőségei



Részösszegek – GROUP BY ROLLUP

Az oszlopneveket és a NULL értéket kombinálva csoportosít, és megjeleníti a részösszegeket valamint végösszeget. Csoportosításkor a nem NULL oszlopok száma jobbról balra csökken

```
SELECT oszlopkifejezések*,
            aggregálás
FROM …
GROUP BY
ROLLUP(oszlopkifejezések*)
```

* Többnyire oszlopnevek listáját jelenti

Pl:

```
SELECT oszlop1, oszlop2, oszlop3,
            SUM(oszlop4)
FROM …
GROUP BY ROLLUP (oszlop1, oszlop2, oszlop3)
```

esetén a következő csoportok jönnek létre:

- Oszlop1, oszlop2, oszlop3
- Oszlop1, oszlop2, NULL
- Oszlop1, NULL, NULL
- NULL, NULL, NULL

Részösszegek – GROUP BY ROLLUP - Példa

RAKTAR_KOD	KAT_ID	MEGYS	Készlet értéke
5	5	db	294000
5	5	NULL	294000
5	NULL	NULL	294000
6	9	db	1262600
6	9	NULL	1262600
6	NULL	NULL	1262600
9	5	db	73000
9	5	NULL	73000
9	9	db	11800
9	9	NULL	11800
9	NULL	NULL	84800
NULL	NULL	NULL	1641400

Példa
Készítsünk listát a raktáron lévő termékek összértékéről raktárkód, azon belül kategóriakód, majd mennyiségi egység szerinti bontásban! A lista jelenítse meg a részösszegeket és a végösszeget is! A listát szűrjük az 5-ös és 9-es azonosítójú kategóriára! A csoportosításnál a ROLLUP záradékot használjuk!

```
SELECT RAKTAR_KOD, KAT_ID, MEGYS,
            SUM(LISTAAR*KESZLET)
             AS 'Készlet értéke' FROM
Termek
WHERE kat_ID IN (5,9)
GROUP BY ROLLUP (RAKTAR_KOD, KAT_ID, MEGYS)
```

Részösszegek – GROUP BY CUBE

Az oszlopneveket és a NULL értéket kombinálva csoportosít, és megjeleníti a részösszegeket valamint végösszeget. A csoportosításhoz minden lehetséges kombinációt felhasznál.

```
SELECT oszlopkifejezések,
            aggregálás
FROM …
GROUP BY
CUBE(oszlopkifejezések)
```

Pl:
SELECT oszlop1, oszlop2, oszlop3, SUM(oszlop4)
FROM …
GROUP BY CUBE (oszlop1, oszlop2, oszlop3) esetén a következő csoportok jönnek létre:

- Oszlop1, oszlop2, oszlop3
- Oszlop1, oszlop2, NULL
- Oszlop1, NULL, oszlop3
- Oszlop1, NULL, NULL
- NULL, oszlop2, oszlop3
- NULL, oszlop2, NULL
- NULL, NULL, oszlop3
- NULL, NULL, NULL

Részösszegek – GROUP BY CUBE- PÉLDA

RAKTAR_KOD	KAT_ID	MEGYS	Készlet értéke
5	5	db	294000
9	5	db	73000
NULL	5	db	367000
6	9	db	1262600
9	9	db	11800
NULL	9	db	1274400
NULL	NULL	db	1641400
NULL	NULL	NULL	1641400
5	NULL	db	294000
5	NULL	NULL	294000
6	NULL	db	1262600
6	NULL	NULL	1262600
9	NULL	db	84800
9	NULL	NULL	84800
5	5	NULL	294000
9	5	NULL	73000
NULL	5	NULL	367000
6	9	NULL	1262600
9	9	NULL	11800
NULL	9	NULL	1274400

Példa
Készítsünk listát a raktáron lévő termékek összértékéről raktárkód, azon belül kategóriakód, majd mennyiségi egység szerinti bontásban! A lista jelenítse meg a részösszegeket és a végösszeget is! A listát szűrjük az 5-ös és 9-es azonosítójú kategóriára! A csoportosításnál a CUBE záradékot használjuk!

```
SELECT RAKTAR_KOD, KAT_ID, MEGYS,
            SUM(LISTAAR*KESZLET)
             AS 'Készlet értéke'
FROM Termek
WHERE kat_ID IN (5,9)
GROUP BY CUBE (RAKTAR_KOD, KAT_ID, MEGYS)
```

GROUPING SETS

A GROUP BY parancs kiegészítve a GROUPING SETS taggal lehetővé teszi, hogy többféle csoportosítást is megadjunk.

A csoportosításokat leíró oszlopkifejezéseket egymás után, zárójelek között , vesszővel elválasztva kell megadni*

Pl:

```
SELECT oszlop1, oszlop2, SUM(oszlop3)
FROM table
GROUP BY
GROUPING SETS(( oszlop1, oszlop2), (oszlop1)
)
```

* A GROUPING SETS-ek az egyes csoportokra kiadott SELECT-ek UNION ALL-jainak alternatívái

GROUPING SETS - PÉLDA

szulev	nem	Átlagos életkor
NULL	F	35
NULL	N	49
1967	NULL	53
1975	NULL	45
1976	NULL	44
1980	NULL	40
1985	NULL	35
1997	NULL	23

Példa
Készítsünk listát az egyes ügyfelek átlagos életkoráról az ügyfél neme, illetve az ügyfél születési éve szerint csoportosítva! A listát szűrjük azon ügyfelekre, akik neve D-vel vagy E-vel kezdődik!
(Az életkor legyen a születési évtől a jelenlegi évig eltelt évek száma)

```
SELECT szulev, nem,
            AVG(YEAR(GETDATE())-SZULEV)
             AS 'Átlagos életkor'
FROM Ugyfel
WHERE NEV LIKE 'E%' OR NEV LIKE 'D%' GROUP BY
GROUPING SETS((SZULEV),(NEM))
```

A GROUPING és GROUPING _ID függvények

A GROUPING fv. értéke 1, ha az adott oszlopkifejezés szerint aggregálás (sum, min, max stb.) van, különben 0

A GROUPING_ID fv. értéke a paraméterként megkapott oszlopkifejezések aggregációs szintjének száma*

- Használhatók a SELECT, a HAVING és az ORDER BY részekben, amennyiben a GROUP BY is specifikálva van
- Mindkét fv. legfontosabb alkalmazása, hogy megkülönböztessük a ROLLUP, CUBE és GROUPING SETS-ek által visszaadott NULL értékeket (részösszegek, végösszegek) a normál NULL értékektől
- Ha egy oszlop(kifejezés) szerint csoportosítunk, akkor a két fv. ugyanazt az értéket adja vissza

* A szintszám számítása úgy történik, hogy az oszloplistához egy bináris kódot rendelünk. Ennek i-ik eleme 1, ha az i-ik oszlop szerint van aggregálás, különben 0. A szintszám a bináris szám értéke lesz tízes számrendszerben. Pl: ha a csoport (oszlop1, oszlop2), és mindkettő szerint van aggregálás, akkor a szintszám binárisan 11, decimálisan 3.

GROUPING & Részösszeg, végösszeg

Pl: listázzuk, hogy melyik évben hány db terméket rendeltek meg! A lista megfelelően jelölve jelenítse meg a rendelések teljes összegét is!

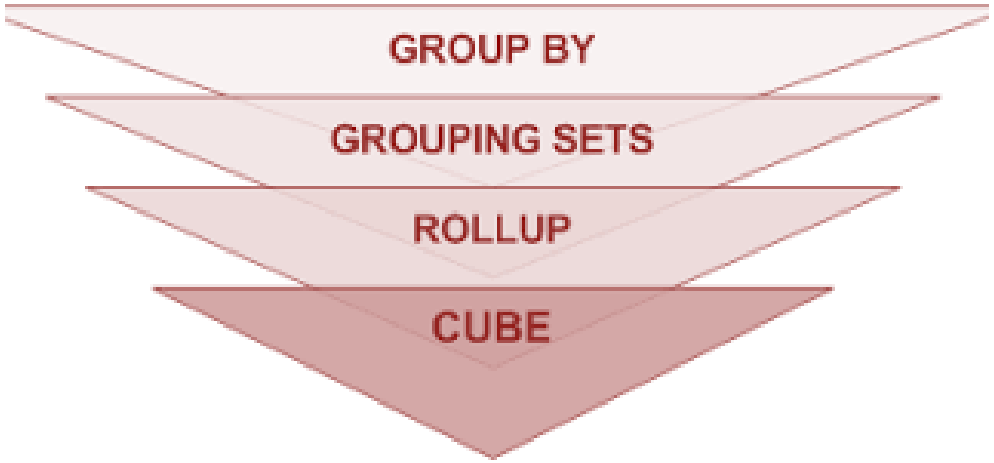
```
SELECT
  (
    CASE GROUPING(YEAR(REND_DATUM))
    WHEN 0 THEN CAST(YEAR(REND_DATUM)
                    AS nvarchar(4))
    WHEN 1 THEN 'Összesen' END
    )
    AS ÉV,
    COUNT(*) AS 'DB'
FROM Rendeles
GROUP BY ROLLUP(YEAR(REND_DATUM))
```

Pl: listázzuk, hogy naponta, azon belül fizetési mód szerint hány rendelés történt!
A lista megfelelően jelölve jelenítse meg a részösszegeket és a végösszeget is!

```
SELECT      IIF(GROUPING(REND_DATUM)=1,'Összesen',
            CAST(REND_DATUM AS nvarchar(10)))
                    AS 'Rendelés dátuma',
  (
    CASE GROUPING_ID(REND_DATUM, FIZ_MOD)
      WHE   0  THEN  FIZ_MOD
        N
      WHE   1  THEN  '**Fizetési módok összesen**'
        N
    END)      as 'FIZ_MOD',
      WHE   3  THEN  'Összesen'
      COUNT(*) as 'DB'
        N
FROM Rendeles
GROUP BY ROLLUP(REND_DATUM, FIZ_MOD)
```

MEGJEGYZÉS

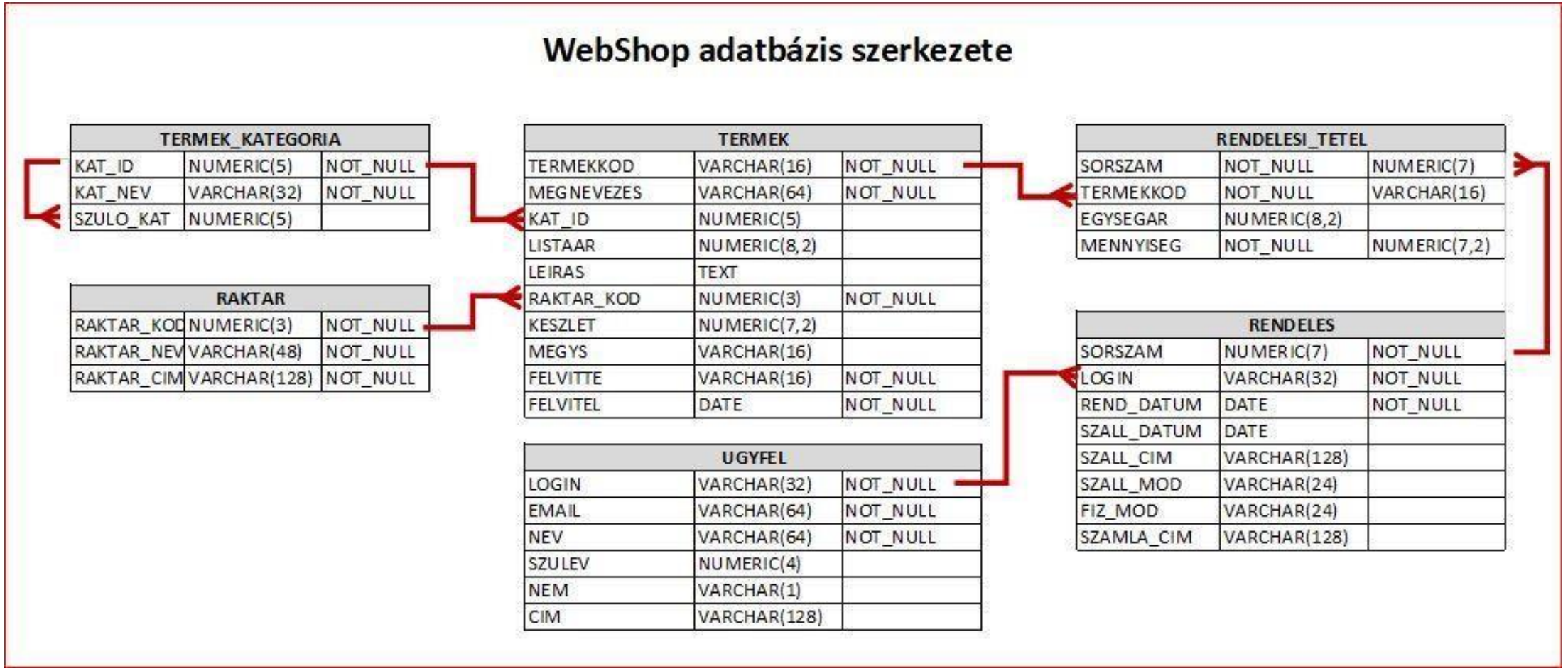
A GROUPING SETS, GROUPING / GROUPING_ID fv.
és a ROLLUP / CUBE együtt is használhatók




```
SELECT CASE
            WHEN GROUPING(szulev)=0
            THEN CAST(SZULEV AS nvarchar(4))
            ELSE 'Minden év'
          END AS 'Születési év', CASE
            WHEN GROUPING(nem)=1
            THEN 'Minden nem'
            ELSE NEM
          END AS 'Nem',
          AVG(YEAR(GETDATE())-SZULEV)
    AS 'Átlagos életkor'
FROM Ugyfel
WHERE NEV LIKE 'E%' OR NEV LIKE 'D%' GROUP BY
GROUPING SETS(ROLLUP(SZULEV),
CUBE(nem))
```

BUDAPESTI CORVINUS EGYETEM

A gyakorlaton használt webshop adatbázis



Köszönöm a figyelmet!