

## Segítségek a 3. gyakorlathoz

### a) feladathoz

A cégnél lévő legkisebb fizetést egy aggregáló lekérdezéssel lekérhetjük, ezt kell beágyazni egy külső SELECT-be.

### b) feladathoz

Az előző lekérdezésből csak a JOB\_ID-t megtartva egy újabb SELECT-be kell beágyazni.

### d) feladathoz

*RPAD* függvény – mező vagy szövegkonstans megjelenítése adott karakterszámon, jobbról feltöltve. Pl.

`RPAD(contry_name, 15)` – az ország neve 15 karakteren, jobbról (alapértelmezetten szóközzel) feltöltve

`RPAD(contry_name, 15, '-')` – a feltöltést kötőjel karakterekkel végzi:  
HUNGARY-----  
ITALY-----

A második paraméter nem csak konstans, hanem egész értékű kifejezés is lehet! Az ezres nagyságrend kiszámítása történhet a `ROUND(min_salary/1000)` kifejezéssel.

A szövegek összefűzését a `||` műveletjel végzi.

### e) feladathoz

A *USING* a *LEFT JOIN* esetén is használható. A rendezés szempontjait sorszámokkal is megadhatjuk (a mezők *SELECT*-beli sorrendjének megfelelően).

**2. feladat:** A felhasználói képernyőre való kiírásra a *DBMS\_OUTPUT.PUT\_LINE()* függvénye alkalmas. A futtatáshoz egy utasításblokkot kell létrehozni, deklarációs rész nélkül, csak *BEGIN* és *END* utasítások közé zárva (2. előadás 24-25. dia).

**3. feladat:**

Változót az *ACCEPT* változónév parancssal kérhet be a felhasználótól. Ha ezt követően egy *PROMPT* 'bekérés szövege: ' parancsot is alkalmaz, akkor a párbeszédablakon megjelenik a bekérés szövege is.

A blokk deklarációs részében két változót kell létrehozni. Az egyiknek a típusa egyezzen meg az *employees.salary* mező típusával (%TYPE segítségével megoldható, 36. és 39. dia). A másik változó – amelyet a létszám megjelenítésére használ – lehet *NUMBER*.

A megoldáshoz nem kell *kurzort* deklarálni, mert ha egyetlen rekordot kérdezzük le (pl. kulcs alapján vagy csoportosítás nélküli aggregálás esetén), akkor az utasításblokkban elhelyezett *SELECT*-ből közvetlenül a változókban küldhetjük a találat mezőinek értékét. Például:

```
SELECT max(salary) INTO maxfizetes FROM employees;
```

A kerekítéshez használhatja a *ROUND* függvényt, ennek második paramétere állítja be a megjelenített tizedes jegyek számát. A szövegek összefűzése a || karakterpárral történik.

**4. feladat:**

A *kurzorok* deklarálásával és kétféle használatával a 2. előadás 38-40. diáján találkozhat.

A felhasználótól bekért változóra a blokkban az & jellel hivatkozhatunk. Szöveg esetén mind a mező tartalmát, mind a felhasználótól kapott sztringet érdemes konvertálni, például:

```
UPPER(department_name) = UPPER('&reszleg')
```

A formázott (fix mezőszélességű) kiíráshoz használhatja az *RPAD* ill. *LPAD* függvényeket.

**5. a) feladat:**

A nézettábla létrehozásakor érdemes a *CREATE OR REPLACE* alakot használni (13. dia), hogy az esetleges ismételt kipróbáláskor ne kelljen mindig törölni a korábbi változatot.

Az összekapcsolást végezheti a *USING* segítségével. A rendezést nem érdemes a nézettáblában végezni. Ha szükséges, akkor majd a lekérdezésben rendez (most szükséges).

Figyelje meg, hogy a *SELECT* eredménye nem kerül a *Dbms Output* ablakba!

**5. b) feladat:**

Mivel van olyan nézettábla, ami tartalmazza a szükséges adatokat, a lekérdezést nem az eredeti táblákból, hanem a nézetből érdemes elkészíteni!

**6. feladat:**

Ha egy *kurzort* módosításra is fel akarunk használni, akkor a deklarációját ki kell egészíteni a *FOR UPDATE* záradékkal (és párhuzamos felhasználásra felkészülve érdemes a *NOWAIT* záradékot is használni; 43-45. dia).

A feldolgozó ciklus formájától függetlenül az esetleges módosítások során a *WHERE CURRENT OF kurzornév* formában tudjuk az éppen feldolgozott rekordot elérni.