

# Adatbázis-kezelő rendszerek II.

pótZH: I. + II. gyakorlati ZH (2021. tavasz), 95 perc, 30+30 pont, 2021. május 27. **10:00-11:35**

- **A ZH megkezdése előtt mindenképp törölje a korábban létrehozott tárolt eljárásait, függvényeit és triggereit a nem kívánatos működés elkerülése végett!!**
- Az alábbi feladatok megoldásához használható a Moodleben lévő jegyzet. Nem használható külső segítség.
- A megoldásokat a Moodlebe kell feltölteni, egyetlen szöveges állomány formájában (<kurzuszsam>\_<NEPTUN>.sql). **Az állomány elején állítsa be a saját adatbázisát alapértelmezettnek (USE <DBNAME>)!**
- A saját adatbázisában dolgozzon! A személygépkocsi (db1) adatbázis tábláit az EXEC [db1]..[sp\_CreateTables] parancs segítségével tudja a saját adatbázisába másolni! Az adatbázis sémája a Moodleben található.
- Aki 11:35-ig nem tölti fel a megoldást, az 11:50-ig megteheti 5 pont levonás mellett. 11:50 után nincs lehetőség a dolgozat beadására.
- Az első és második gyakorlati ZH pótlása esetén a pontszámok A+B formában jelennek meg, ahol A az I. ZH-ba, B pedig a II. ZH-ba számító pontszámot jelenti.

**Beadandó:** a lenti feladatok T-SQL scriptjei egyetlen szöveges (.sql) állományban. Megjegyzések, tesztelési adatok, tájékoztatók kommentben szerepeljenek. Hibára futó kódrészlet legyen kikommentezve megjegyzéssel! Ha egy komment nélküli kódrész nem fut le, vagy megjegyzés nélkül van kommentezve, úgy nem kerül kiértékelésre.

## 1. feladat

**1A:** Készítsen olyan SQL lekérdezést, mely a kölcsönzés éve és az ügyfél országazonosítója alapján vett bontásban adja vissza a kölcsönzött kocsik árának átlagát (szgk.ar)!

**(5+0 pont)**

**1B:** Alkalmazzon PIVOT vagy UNPIVOT lekérdezést az előző lekérdezés eredménytáblájára. PIVOT lekérdezés esetén, ha több érték is rendelkezésre áll, úgy azok maximumát adja vissza!

**(3+0 pont)**

## 2. feladat

**2A:** Készítsen olyan **felhasználói függvényt** a saját adatbázisba, melynek **1 paramétere** van: egy kölcsönzés azonosítója. A függvény ellenőrizze le, hogy helyesen van-e kiszámolva a kölcsönzési díj (kolcsdij = ar \* eltelt napok száma, eltelt napok száma = hatdat és kidat alapján napokban, az ar attribútum az szgk táblában található). Ha nem létezik az adott kölcsönzés, akkor a visszatérési érték NULL. Ha létezik, akkor a visszatérési érték a valós kölcsönzési díjtól való abszolút eltérés (pl: adatbázisban: 5 vagy 9, valós érték: 7 -> eredmény: 2)!

**(5+5 pont)**

**2B:** Tesztelje a függvényt!

**(0+2 pont)**

### 3. feladat

Készítsen egy tárolt eljárást, amelynek paramétere egy ügyfélazonosító:

- Az eljárás először ellenőrzi, hogy létezik-e a paraméterben megkapott azonosítójú ügyfél. Ha nem, akkor ezt kiírja, és a tárolt eljárás fejezze be a futását! **(3+3 pont)**
- Ha van ilyen ügyfél, akkor ellenőrizze, hogy az ügyfélnek van-e olyan kölcsönzése, ahol a kölcsönzési díj hibás? Ehhez használja az első feladatban megírt függvényt! Ha nincs, úgy a rendszer tájékoztassa a felhasználót erről és az eljárás fejezze be a futását! **(0+2 pont)**
- Ha van(nak) hibás kölcsönzése(i), úgy frissítse a kölcsönzések táblának az ügyfélhez tartozó hibás rekordokat a megfelelő összegre! **(0+2 pont)**
- Az eljárás tranzakciót futtat. Ha bármilyen hiba történik, az eljárás kiírja a hibakódot és a rendszertől származó angol nyelvű hibaüzenetet. **(3+3 pont)**
- Tesztelje le az összes ágot, a teszteléshez használt paramétereket is tüntesse fel! **(4+4 pont)**
- Állítsa vissza az eredeti állapotot. **(2+2 pont)**

### 4. feladat

**4A:** Készítsen egy **trigger**t, amely a kölcsönzéses tábla frissítésekor fut le. A trigger írja ki (PRINT) a következő adatokat: kölcsönző azonosítója, ügyfél neve, kölcsönzött autó rendszáma, régi kölcsönzési díj, új kölcsönzési díj. **(5+5 pont)**

**4B:** Tesztelje a trigger a 2. feladat tárolt eljárásának segítségével! **(0+2 pont)**