

# Informatikai technológiák laboratórium 1.

## Hallgatói segédlet

## Az Entity Framework programozása

### Mérés célja

A mérés célja, hogy a hallgatók bemutassák, és elmélyítsék az Entity Framework programozása területén szerzett ismereteiket. A labor ennek megfelelően részben vezetett, részben önállóan végrehajtandó feladatokból épül fel.

### Szükséges ismeretek

- Az SQL nyelv ismerete (DML, SELECT utasítás).
- TS-QL programozási ismeretek tárolt eljárások létrehozásához
- C# programozási alapismeretek
- C# Language Integrated Query (LINQ) ismerete
- Entity Framework ismeretek
- Visual Studio valamint az SQL Server Management Studio ismerete a feladatok megoldásához.

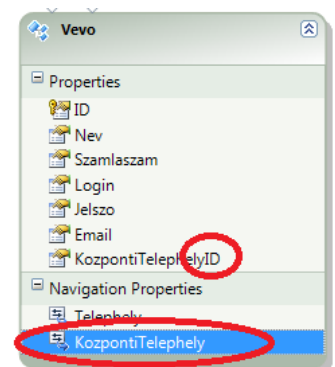
### Egyéb információk a méréshez

- A mérés során SQL Server localdb adatbázissal kell dolgozni.
- A szerver elérhetőségét és a felhasználói információkat a mérésvezető az óra elején elmondja.
- A mérés során elvégzett feladatok dokumentálására jegyzőkönyvet kell készíteni.

### Mérési feladatok

1. Indítsd el a Visual Studio 2013-at, hozz létre egy új konzolos alkalmazást (.NET 4.0, C#). A labor során ezt az alkalmazást fogjuk bővíteni. Adj hozzá egy új Entity Data Model-t a projecthez, a modell tartalmazza a mintaadatbázis összes tábláját! A további feladatok megoldásához (ha a feladat nem rendelkezik máshogy) az így legenerált osztályokat használd! (Op.)
2. Készíts egy függvényt `TermekListazas()` néven, mely kiírja a termékek neveit (2p.)

3. Készíts egy előzőhöz hasonló függvényt `HatekonyTermekListazas()` néven, mely csak a készleten lévő termékeket írja ki, a termékeket betűrendbe rendezi, illetve hogy az adatbázisból csak a megjelenítendő adatokat kéri le! (összesen 6p.)
4. Készíts egy függvényt `TermekListazasBrutto()` néven mely a termékek neveit és bruttó árait írja ki. (4p.)
5. Készíts egy függvényt `Megrendelesek()` néven, mely felsorolja a termékeket és minden termék mellett feltünteti a megrendelések összegét! Ügyelj rá, hogy a listázás egyetlen SQL utasítás végrehajtásával történjen (összesen: 6p.)
6. Készíts függvényt `Eladas()` néven mely a 3-as azonosítójú termék készletét 1-gyel csökkenti, majd kiírja az új készletet. (4p.)
7. Készíts egy függvényt `Beszallitas()` néven, mely felvesz egy új kategóriát és egy abba tartozó új terméket az adatbázisba, végül pedig kiírja az új sorok azonosítóit. A művelet során ne használjuk ki, hogy ismerhetjük az új kategória azonosítóját (összesen 6p.)
8. Az Entity Framework segítségével válaszd szét az ÁFA mentes számlatételket a többtől osztály szinten is. Ennek érdekében vezess be egy `AfaMentesSzamlaTetel` entitástípust, mely a `SzamlaTetel` típussal származási relációban van! Megoldásod tesztelésére készíts egy függvényt `AfaMentesTetelek()` néven, mely kilistázza az összes számlatételt, megjelölve azok osztályát! (Tipp: az alapértelmezett adatbázisban nincs 1-es kódú tétel, a megfelelő teszteléshez új számlatételeket kell felvenned, vagy meglévőket módosítani. Ezt megteheted közvetlenül az SQL Server Management Konzol használatával is.) (10p.)
9. Készíts egy függvényt `VevoListazas()` néven, mely kiírja a vevők neveit (4p.)
10. Készíts egy előzőhöz hasonló függvényt `HatekonyVevoListazas()` néven, mely csak az „o” betűt tartalmazó nevű vevőket listázza ki, a vevőket betűrendbe rendezi, illetve az adatbázisból csak a megjelenítendő adatokat kéri le! (összesen 6p.)
11. Készíts egy függvényt `VevoTelephelyListazas()` néven mely a vevők neveit és központi telephelyeinek címeit listázza ki. Az EDMX varázsló a KözpontiTelephelyhez tartozó tulajdonságokat nem tudja automatikusan jól elnevezni. Az egyszerűbb munka érdekében módosítsd az edmx szerkesztőben a `KozpontiTelephely` tulajdonság nevét `KozpontiTelephelyID`-re, majd a `Telephely1` navigációs tulajdonságot `KozpontiTelephelyre` (6p.)
12. Készíts egy függvényt `VevoMegrendelesek()` néven, mely felsorolja a vevőket és minden vevő mellett feltünteti az eddigi, központi telephelyről feladott megrendeléseinek számát! Ügyelj rá, hogy a listázás egyetlen SQL utasítás végrehajtásával történjen (összesen: 10p.)
13. Készíts függvényt `Rendeles()` néven, mely a Puskás Norbert nevű vevő számára minden termékből megrendel egyet. (A megoldás itt értelemszerűen több lekérdezést is tartalmazhat) (14p.)



14. Készíts egy függvényt `Kiszallitas()` néven, mely minden megrendelés státuszát 5-re („Kiszállítva”) állítja. A módosítás során a megrendelés tételeinek a státuszát is módosítani kell. (összesen 6p.)
15. Egészítsd ki a `Vevo` osztályt egy `Keresztnev` tulajdonsággal mely a kéttagú „Nev” második tagját adja vissza. Munkádat teszteld a `VevoNevek` függvényben (Tipp: a `Vevo` osztály parciális). (6p.)
16. Válaszd szét a sztornózott és a nem sztornózott számlákat entitás szinten! Használd az EF lezármaztatási lehetőségeit! Megoldásod szemléltetésére készíts egy `Szamlak()` nevű függvényt, mely kilistázza a rendszerben tárolt számlákat (ID és összeg), megjelölve azok típusát! (10p.)

## Ellenőrző kérdések

- Sorold fel az Entity Framework három leképzési rétegét!
- Milyen lehetőségek vannak Entity Frameworkben egy osztály adattáblákra való leképzésére?
- Mit jelent a „lazy loading”, melyen szerepet kap az Entity Frameworkben?
- Az Entity Framework 3 alapvető módszert kínál az entitások elérésére. Melyek ezek?
- Miért jelentenek nehézséget az ORM rendszerek szempontjából a 3, vagy többretegű alkalmazások a hagyományos 2 rétegű alkalmazásokkal szemben?
- Milyen problémára és hogyan ad választ a „self-tracking entity” megoldás az Entity Frameworkben?
- Milyen problémára és hogyan adnak választ a `POCO` és a `POCO proxy` osztályok az Entity Frameworkben?