Fodrászat

# Tartalom

[Tartalom 2](#_Toc162014427)

[Bevezetés 4](#_Toc162014428)

[Miért ezt választottuk? 4](#_Toc162014429)

[Rövid megfogalmazás 4](#_Toc162014430)

[Előnye 4](#_Toc162014431)

[Megjegyzés 4](#_Toc162014432)

[Szerkezete 4](#_Toc162014433)

[Egyedek (és attribútumok) 4](#_Toc162014434)

[Vendég 4](#_Toc162014435)

[Fodrász 5](#_Toc162014436)

[Üzlet 5](#_Toc162014437)

[Szolgáltatás 5](#_Toc162014438)

[Helyszín 5](#_Toc162014439)

[Kapcsolatok 5](#_Toc162014440)

[Üzlet-helyszín 5](#_Toc162014441)

[Foglalás 5](#_Toc162014442)

[Egyed-kapcsolat diagram (ER-modell) 7](#_Toc162014443)

[Relációs adatmodell 8](#_Toc162014444)

[Normálformák 8](#_Toc162014445)

[0. Normálformában 8](#_Toc162014446)

[1. Normálformában 9](#_Toc162014447)

[2. Normálformában 10](#_Toc162014448)

[3. Normálformában 11](#_Toc162014449)

[Kapcsolat-modell 12](#_Toc162014450)

[Teszt adatok generálása 13](#_Toc162014451)

[Cél és Áttekintés 13](#_Toc162014452)

[Fő Program 13](#_Toc162014453)

[Fontos Megjegyzések: 13](#_Toc162014454)

[Használati Példa: 13](#_Toc162014455)

[Funkciók 14](#_Toc162014456)

[CimBe(): 14](#_Toc162014457)

[SzolgBe(): 14](#_Toc162014458)

[kiszolgalox(): 14](#_Toc162014459)

[NevGeneralas(int db, bool kellemail, List<string> lista): 14](#_Toc162014460)

[VnevKnevBe(): 14](#_Toc162014461)

[TelefonszamKigeneralas(): 14](#_Toc162014462)

[VaneIlyen(string ujsz)**¹**: 14](#_Toc162014463)

[Feladatok 15](#_Toc162014464)

[Könnyű feladatok 15](#_Toc162014465)

[Közepes feladatok 17](#_Toc162014466)

[Nehéz feladatok 18](#_Toc162014467)

[Jegyzetek és források 19](#_Toc162014468)

[Forráskód és gyorslinkek 19](#_Toc162014469)

[Online Repository Link 19](#_Toc162014470)

[Source kód(Generálás) 19](#_Toc162014471)

[Source kód(Tábla szerkezet) 19](#_Toc162014472)

[Diagramok 19](#_Toc162014473)

[Használt programmok 19](#_Toc162014474)

[XAMPP 19](#_Toc162014475)

[Visual Studio Code 19](#_Toc162014476)

[Visual Studio 19](#_Toc162014477)

[Notepad++ 19](#_Toc162014478)

[Word 19](#_Toc162014479)

# Bevezetés

## Miért ezt választottuk?

Úgy gondoljuk, minden fodrászat rendelkezik egy adatbázissal, egy általános példáját mutatjuk be ebben a dokumentációban.

## Rövid megfogalmazás

Egy fodrászat adatbázisa egy strukturált adathalmaz, amely tartalmazza a fodrászat **ügyfeleinek**, **alkalmazottainak** és **tevékenységeinek** információit. Az adatbázisban szerepelnek az **ügyfelek nevei, elérhetőségei és korábbi kezeléseikkel kapcsolatos információk**, mint például a kiválasztott **szolgáltatások és annak időpontja**. Emellett tárolja az **alkalmazottak adatait** is, beleértve **nevüket és elérhetőségüket**. Külön tárolja a **szolgáltatások árát, nevét és időtartalmát is**. Az adatbázis az **üzleteket** is kezeli, mivel **azok rendelkezhetnek leányvállalatokkal,** így helyszínüket is külön tároljuk. Feltételezzük, hogy ez egy olyan fodrászat, ahova nem kell időpontot foglalni.

## Előnye

A fodrászat adatbázisa hatékony eszköz a mindennapok megkönnyebbítésére és akár **tudja javítani a kommunikációt és az üzlet teljesítményét is**.

## Megjegyzés

A következő oldalon röviden olvashat egy ilyen adatbázis szerkezetéről és **egyed-kapcsolat modell** (vagy **ER-modell**) formájában is megtekintheti.

# Szerkezete

## Egyedek (és attribútumok)

### Vendég

* + Telefonszám**[[1]](#footnote-1)** – Elsődleges kulcs. (PK)
  + Név – A vendég neve.
  + Email – A vendég email címe.

### Fodrász

* + Telefonszám**[[2]](#footnote-2)** – Elsődleges kulcs. (PK)
  + Név – Az alkalmazott neve.
  + Email – Az alkalmazott email címe.

### Üzlet

* + Üzletvezető3 – Az üzlet vezetője.
  + id. – Elsődleges kulcs. (PK)

### Szolgáltatás

* + Név – A szolgáltatás neve, egyben az elsődleges kulcs is. (PK)
  + Idő – A szolgáltatás hossza, nem időpontja. (Perc)
  + Ár – A szolgáltatás ára. (Ft)

### Helyszín

* + Cím – A helyszín címe (Utca, házszám stb.…).
  + Irányitószám – A város irányítószáma, amiben az üzlet helyezkedik el.
  + id. – Elsődleges kulcs. (PK)

## Kapcsolatok

### Üzlet-helyszín

Az üzlet-helyszín kapcsolat egy **áthidaló elem két egyed között. Több leányvállalat létezhet**, **több városban,** ezért kénytelennek vagyunk a **helyszínt kiszervezni** egy külön egyedként. A kapcsolat **az összekötéshez szükséges**.

### Foglalás

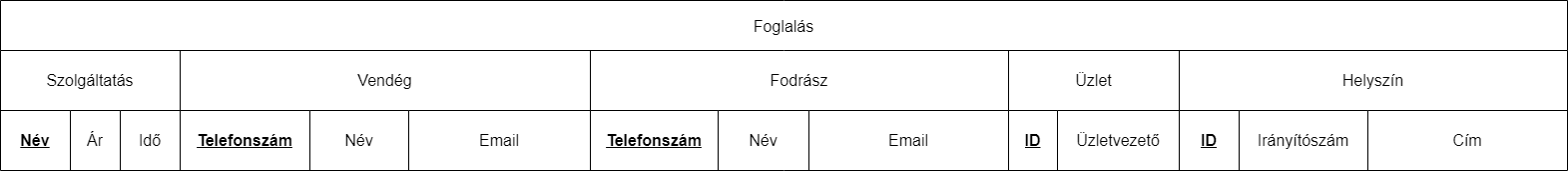
A foglalás kapcsolat, **egy összefűző része** az adatbázisnak. Lényegében **összeköti a fodrászokat a vendégekkel**. A kapcsolat **tárolja** még a **szolgáltatást és az üzletet**, amiben a kezelés történik.

## H:\IKT projektmunka\11. IKT\Adatbázisos projekt\ER-diagram\adatbazis.pngEgyed-kapcsolat diagram (ER-modell) [[3]](#footnote-3)

# Relációs adatmodell

## Normálformák

### Normálformában



### Normálformában

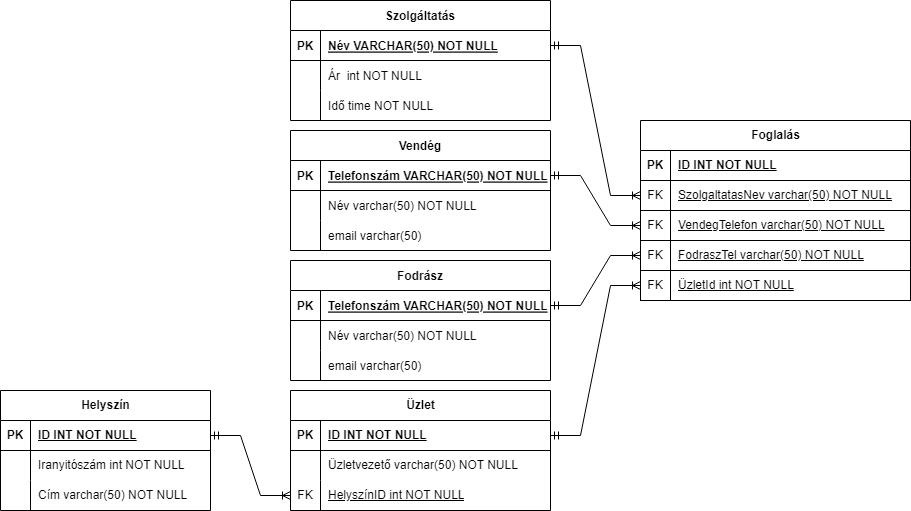


### Normálformában



### H:\IKT projektmunka\11. IKT\Adatbázisos projekt\ER-diagram\3normalforma.pngNormálformában

## Kapcsolat-modell



# Teszt adatok generálása

## Cél és Áttekintés

A "Generálás" program egy adatgeneráló alkalmazás, amely különböző adatokat generál, majd ezeket felhasználva SQL fájlt készít egy fodrászat adatbázis inicializálásához.

## Fő Program

A **Main** metódusban először beolvassuk a fájlokat, majd kigenerálunk véletlenszerű adatokat. Végül ezek az adatok kerülnek beillesztésre egy SQL fájlba az **SqlFajlKi()** metódus segítségével.

## Fontos Megjegyzések:

A programban **használt fájlok elérési útvonala relatív**, tehát a program futási mappájában kell lenniük. A kimeneti **SQL fájl neve** **"fodraszat\_adatok.sql**".

## Használati Példa:

1. Először biztosítsa, hogy a programhoz szükséges fájlok (**cimek.txt, szolgaltatasok.txt, kiszolgalok.txt, knev.txt, vnev.txt**) a futási mappában legyenek.
2. Futtassa a programot.
3. Az alkalmazás a generált adatokat bele fogja írni a "**fodraszat\_adatok.sql**" fájlba.

## Funkciók

### CimBe():

Betölti a "cimek.txt" fájlból az üzletek címeit a "címek" listába.

### SzolgBe():

Betölti a "szolgaltatasok.txt" fájlból a szolgáltatásokat a "szolgaltatasok" listába.

### kiszolgalox():

Betölti a "kiszolgalok.txt" fájlból a kiszolgálók neveit a "kiszolgalok" listába.

### NevGeneralas(int db, bool kellemail, List<string> lista):

Generál véletlenszerű neveket a megadott mennyiségben és hozzárendel egy e-mail címet a kiszolgálók lista segítségével, ha a "kellemail" paraméter igaz.

### VnevKnevBe():

Betölti a "knev.txt" és "vnev.txt" fájlokból a keresztneveket és vezetékneveket a megfelelő listákba.

### TelefonszamKigeneralas():

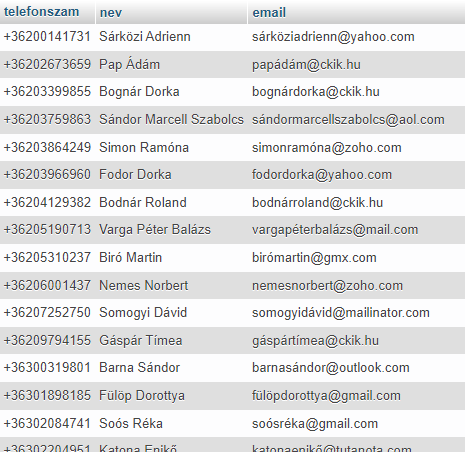
Generál véletlenszerű telefonszámokat**¹** és ezeket a **telefonszamok** listába menti.

### VaneIlyen(string ujsz)**¹**:

Ellenőrzi, hogy az adott telefonszám már létezik-e a listában.

# Feladatok

## Könnyű feladatok

1. Írassa ki az összes fodrász adatát!

2. Egy új fodrász érkezett az adatbázisba. Illeszd be az adatait a fodrász táblába! (tel: +60 124 4522, neve: Betyár Tamás, email: betyarvagyok@gmail.com)

3. Megszeretnénk nézni az összes ceglédi üzlet címét. Írasd ki az összes ilyen címet a helyszín táblából!

4. Kíváncsiak vagyunk arra, hogy a vendégek közül kik azok, akik a közgázba járnak. Írasd ki a diákok nevét!

5. Sándor Marcell Szabolcs nevű fodrász email címet változtatott. Frissítsd adatait! Új email címe: lonelybarber@outlook.com

## Közepes feladatok

1. Írassa ki azok fodrászok nevét, Akik 20-as telefonszámot használnak!

2. Sorold be a szolgáltatásokat kategóriába! Amelyik 5000 forint alatt van az legyen „egyszerű kezelés”, amik felette pedig „összetett kezelés”, **ha nem kerül semmibe (0) akkor legyen üres**! Ha nem tudjuk az árt akkor legyen „?”! Végképp, ha pont 5000 forintba kerül a szolgáltatás az „EO kezelés” jelenjen meg!



## Nehéz feladatok

1. Adja meg 06701923124 telefonszámú fodrász összes foglalását és további adatait is!

2. Írassa ki azokat a fodrászokat, akiknek egyik vendégje 2 keresztnévvel rendelkezik! Minden adat legyen egyedi és ABC sorrendbe legyen rendezve!

3. Írassa ki azokat a foglaláskat ahol a vendég telefonszáma és a fodrász telefonszáma ugyan annál a szolgáltatónál van!

# Jegyzetek és források

## Forráskód és gyorslinkek

### [Online Repository Link](https://github.com/PetrooAdam/fodraszatprojekt)

### [Source kód(Generálás)](https://raw.githubusercontent.com/PetrooAdam/fodraszatprojekt/main/C%23/Gener%C3%A1l%C3%A1s/Gener%C3%A1l%C3%A1s/Program.cs)

### [Source kód(Tábla szerkezet)](https://raw.githubusercontent.com/PetrooAdam/fodraszatprojekt/main/adatbazis/fodraszat_adatbazis.sql)

### [Diagramok](https://github.com/PetrooAdam/fodraszatprojekt/tree/main/ER-diagram)

## Használt programmok

### [XAMPP](https://www.apachefriends.org/hu/index.html)

### [Visual Studio Code](https://code.visualstudio.com/)

### [Visual Studio](https://visualstudio.microsoft.com/)

### [Notepad++](https://notepad-plus-plus.org/downloads/)

### Word

1. Feltételezzük azt, hogy **minden vendég egy meghatározó telefonszámmal rendelkezik**, mivel amikor regisztrálnak egy fodrászatba, csak **egyet** kell megadni. Több megadása **irracionális**. Egy **névből** **több létezhet,** ezért **nem lehet elsődleges kulcs**. [↑](#footnote-ref-1)
2. Az **előbb1** említettek alapján, **egy meghatározó telefonszámmal rendelkezik**.

   3 Feltételezzük, hogy egy üzletvezetőhöz több cím is tartozhat, mivel több üzletet is vezethet. [↑](#footnote-ref-2)
3. Az **egyed-kapcsolat modell** adott tudásterületen belüli, **egymással összefüggő dolgokat ír le**. Egy alapvető ER-modell egyedtípusokból áll és meghatározza az egyedek között létező kapcsolatokat. **Ez egy vizuális formája**. [↑](#footnote-ref-3)