Fodrászat

Tartalom

[Bevezetés 4](#_Toc161168020)

[Miért ezt választottuk? 4](#_Toc161168021)

[Rövid megfogalmazás 4](#_Toc161168022)

[Előnye 4](#_Toc161168023)

[Megjegyzés 4](#_Toc161168024)

[Szerkezete 5](#_Toc161168025)

[Egyedek (és attribútumok) 5](#_Toc161168026)

[Vendég 5](#_Toc161168027)

[Fodrász 5](#_Toc161168028)

[Üzlet 5](#_Toc161168029)

[Szolgáltatás 6](#_Toc161168030)

[Helyszín 6](#_Toc161168031)

[Kapcsolatok 7](#_Toc161168032)

[Üzlet-helyszín 7](#_Toc161168033)

[Foglalás 7](#_Toc161168034)

[Egyed-kapcsolat diagram (ER-modell) 8](#_Toc161168035)

[Relációs adatmodell 9](#_Toc161168036)

[Normálformák 9](#_Toc161168037)

[0. Normálformában 9](#_Toc161168038)

[1. Normálformában 10](#_Toc161168039)

[2. Normálformában 11](#_Toc161168040)

[3. Normálformában 12](#_Toc161168041)

[Kapcsolat-modell 13](#_Toc161168042)

[Teszt adatok generálása 14](#_Toc161168043)

[Cél és Áttekintés 14](#_Toc161168044)

[Fő Program 14](#_Toc161168045)

[Fontos Megjegyzések: 14](#_Toc161168046)

[Használati Példa: 14](#_Toc161168047)

[Funkciók 15](#_Toc161168048)

[CimBe(): 15](#_Toc161168049)

[SzolgBe(): 15](#_Toc161168050)

[kiszolgalox(): 15](#_Toc161168051)

[NevGeneralas(int db, bool kellemail, List<string> lista): 15](#_Toc161168052)

[VnevKnevBe(): 15](#_Toc161168053)

[TelefonszamKigeneralas(): 16](#_Toc161168054)

[VaneIlyen(string ujsz)**¹**: 16](#_Toc161168055)

[Jegyzetek és források 17](#_Toc161168056)

[Forráskód és gyorslinkek 17](#_Toc161168057)

[Online Repository Link 17](#_Toc161168058)

[Source kód(Generálás) 17](#_Toc161168059)

[Source kód(Tábla szerkezet) 17](#_Toc161168060)

[Diagramok 17](#_Toc161168061)

[Használt programmok 17](#_Toc161168062)

[XAMPP 17](#_Toc161168063)

[Visual Studio Code 17](#_Toc161168064)

[Visual Studio 17](#_Toc161168065)

[Notepad++ 17](#_Toc161168066)

[Word 17](#_Toc161168067)

# 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 **az üzletek 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

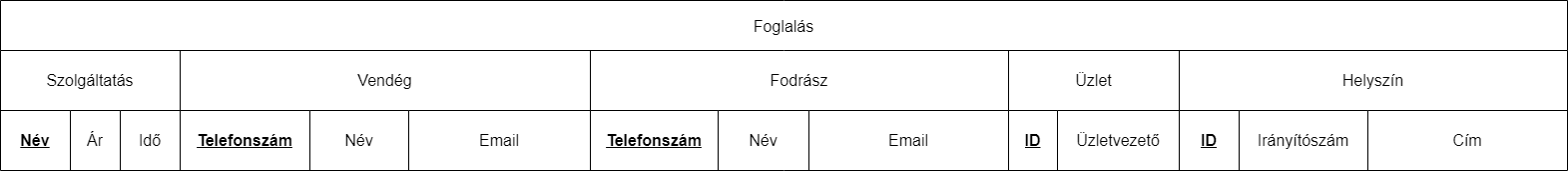
A foglalás kapcsolat, **egy összefűző része** az adatbázisnak. Lényegében **összeköti a fodrászokat a vendégekkel**. A kapcsolat **tárolja** még a **szolgáltatást és az üzletet**, amiben a kezelés történik.

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

# Relációs adatmodell

## Normálformák

### Normálformában



### Normálformában

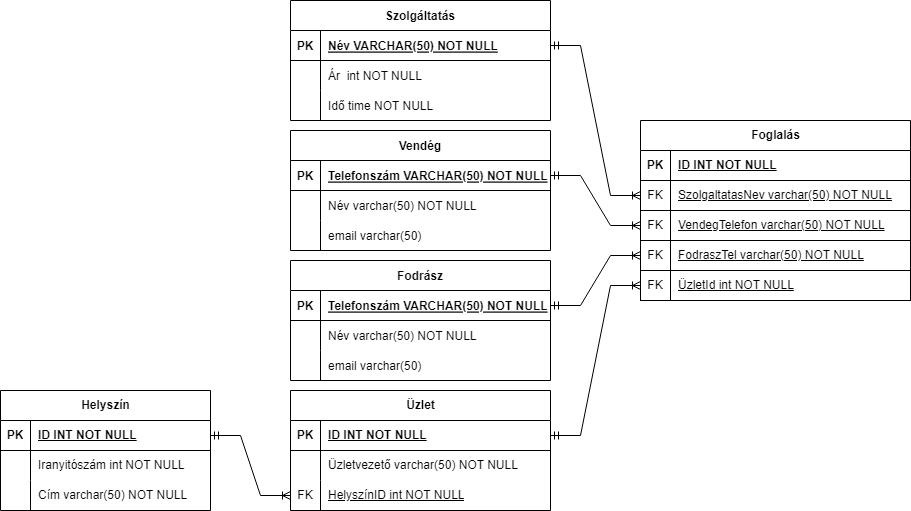


### Normálformában



### 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 ki generálunk 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 hozzárendeli hozzájuk egy e-mail címet a kiszolgálók listából, 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.

# 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 **egy telefonszámot** kell megadni. Több megadása **irracionális**. Egy **névből** **több létezhet,** ezért **nem lehet elsődleges kulcs**. [↑](#footnote-ref-1)
2. Az **előzőbben1** említett 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)