

# JEGYZŐKÖNYV

## Adatbázis rendszerek I.

### Gyakorlat 1

Készítette: Tótok Tamás

Neptunkód: ZY7596

Project link: [GitHub](#)

Dátum: 2025.04.03.

## 1. Feladat

C nyelven készítsen egy programot, amely szabvány billentyűzetről olvasson be sorokat, egészen a végjelig (#). A beolvasott szöveg: a hallgató teljes neve és a neptunkód. A beolvasott sorokat írja ki egy szövegfájlba (neptunkod.txt). A szövegfájl nevét a bevitel első sorában adja meg (File name). Az így létrehozott, lezárt állományt utána nyissa meg és írja vissza a lementett szöveget nagybetűs formában.

**Mentés:** Neptunkod\_1.c

## Megoldás

A szöveges állomány nagybetűs formába alakításhoz létrehozunk egy segédfüggvényt:

```
void to_upper(char *str)
{
    while (*str)
    {
        *str = toupper((unsigned char)*str);
        str++;
    }
}
```

Beállítjuk a karakterkódolást egy újabb segédfüggvény segítségével:

```
void set_encoding()
{
    system("chcp 65001 > nul");
    setlocale(LC_ALL, "hu_HU.UTF-8");
}
```

Bekérjük az adatokat:

```
void read_user_data(const char *prompt, char *buffer, size_t size,
const char *data_name)
{
    do
    {
        printf("%s", prompt);
        fgets(buffer, size, stdin);
        buffer[strcspn(buffer, "\n")] = '\0';

        // Ellenőrizzük, hogy a beírt adat végén # szerepel-e
        if (buffer[strlen(buffer) - 1] != '#')
        {
            printf("Error: Your %s must end with a '#' character.
Please try again!\n", data_name);
        }
    } while (buffer[strlen(buffer) - 1] != '#');
```

```

// Ha az adat végén # szerepel, akkor eltávolítjuk azt
buffer[strlen(buffer) - 1] = '\\0';
}

```

Létrehozzuk a fájlt a megadott névvel:

```

FILE *open_file(const char *filename, const char *method)
{
    FILE *file = fopen(filename, method);

    if (!file)
    {
        perror("Opening file error!");
        exit(1);
    }

    return file;
}

```

Elmentjük a bekért adatokat a létrehozott fájlban:

```

void save_data_to_file(FILE *file, const char *fullname, const char
*neptun_code)
{
    fprintf(file, "%s\\n", fullname);
    fprintf(file, "%s\\n", neptun_code);
    fclose(file);
}

```

Kiírjuk a fájl tartalmát nagybetűs formában:

```

void print_file_text_upper(FILE *file)
{
    char line[MAX_LENGTH];
    printf("\\n");
    while (fgets(line, MAX_LENGTH, file))
    {
        to_upper(line);
        printf("%s", line);
    }
    fclose(file);
}

```

A függvényeket sorban lefuttatjuk a main függvényben:

```
int main()
{
    char filename[MAX_LENGTH];
    char fullname[MAX_LENGTH];
    char neptun_code[20];
    char txt_filename[MAX_LENGTH + 4];

    // Karakterkódolás beállítása
    set_encoding();

    // Adatok beolvasása
    read_user_data("File name: ", filename, sizeof(filename), "file
name");
    read_user_data("Full name: ", fullname, sizeof(fullname), "full
name");
    read_user_data("Neptun code: ", neptun_code, sizeof(neptun_code),
"neptun code");

    // .txt kiterjesztés hozzáadása az általunk megadott fájlhoz
    snprintf(txt_filename, sizeof(txt_filename), "%s.txt", filename);

    // Fájl megnyitása és adatok mentése
    FILE *file = open_file(txt_filename, "w");
    save_data_to_file(file, fullname, neptun_code);

    // Fájl olvasása és tartalom kiírása nagybetűsen
    file = open_file(txt_filename, "r");
    print_file_text_upper(file);

    // Kilépés előtt várunk egy Enter lenyomásra
    printf("\nPress Enter to exit...");
    getchar();

    return 0;
}
```

## 2. Feladat

a.) Készítse el az ER modellt a Termékek és Gyártók leírására. Minden Termék csak egy Gyártóhoz tartozik. A Termék a (Tkód, név, ear), a Gyártó a (Adószám, név, telephely) jellemzi.

Mentés: Neptunkod\_2a.draw.io

### a.) Megoldás

2a:

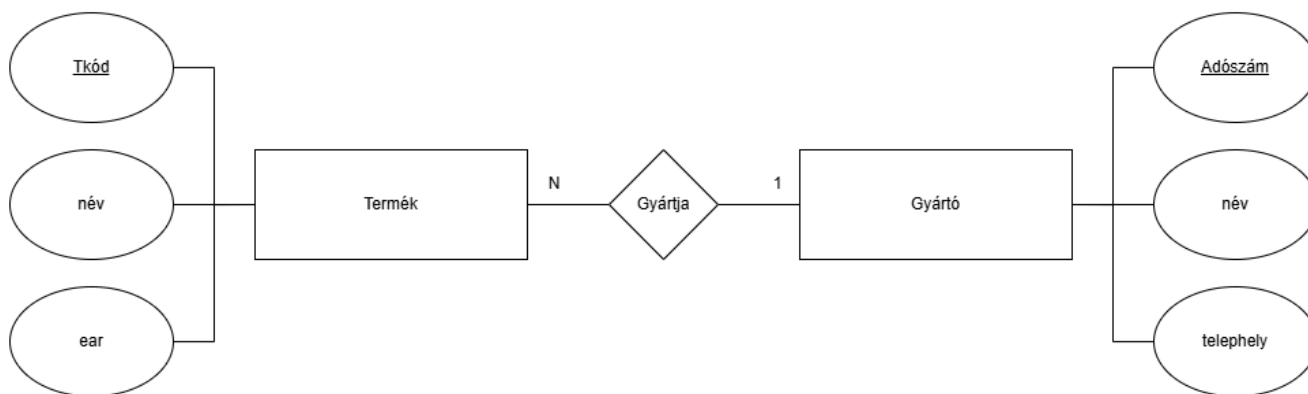
Létrehozzuk a **Termék** és a **Gyártó** egyedeket a következő attribútumokkal:

**Termék:** Tkód, név, ear

**Gyártó:** Adószám, név, telephely

Végül a két egyedet összekapcsoljuk a megfelelő kapcsolattípussal:

**Termék N—1 Gyártó**



b.) Az 1. feladatban elkészített ER modellt bővítse a következőképpen: - postai cím a gyártóhoz megadva a cím komponenseit külön-külön is, - egy Terméknél a kapcsolódó csomagolási egységek darabszámai - több ilyen egység is lehet, - a Termék és az Alkatrész (Akód, név) adatait, ahol egy Alkatrész több Termékben is megjelenhet és fordítva is.

Mentés: Neptunkod\_2b.draw.io

### b.) Megoldás

2b:

A **Gyártó** egyedhez hozzáadjuk a postai cím összetett attribútumot:

**Gyártó:** Adószám, név, telephely, postai cím (irányítószám, város, utca, házszám)

Létrehozzuk az **Alkatrész** és a **Csomagolási egység** nevű egyedeket:

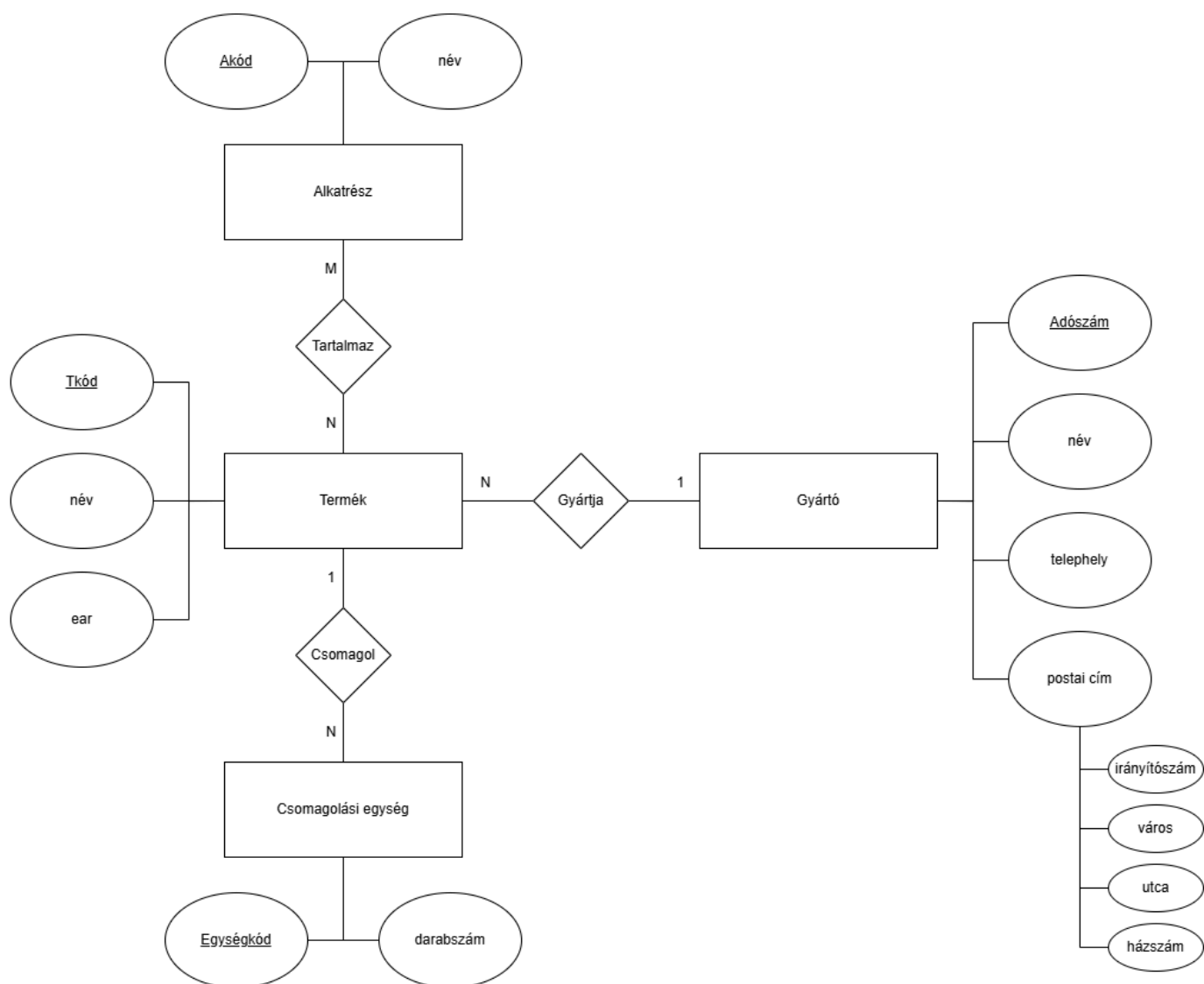
**Alkatrész:** Akód, név

**Csomagolási egység:** Egységkód, darabszám

Végül a három egyedet összekapcsoljuk a következőképpen:

**Alkatrész M—N Termék**

**Termék 1—N Csomagolási egység**



### 3. Feladat

Az 2b. feladatnál létrehozott Termék-Gyártó modellt bővítse a következőképpen: a. A Termék-hez a Rendelés-ek (egyed) nyilvántartása, amelynél egy termékből többben is rendelhetnek és fordítva is. A Rendelés-nél a Rkód, Dátum, Darabszám, és Ár, amely darabszám \* ear formával számolódik. A Vevő a Rendeléssel van kapcsolatba, úgy, hogy egy Vevő többször is Rendelhet. A vevő tulajdonságai Vkód, név, lakcím összetett tulajdonság.

**Mentés:** neptunkod\_3.draw.io

## Megoldás

Létrehozzuk a kért egyedeket:

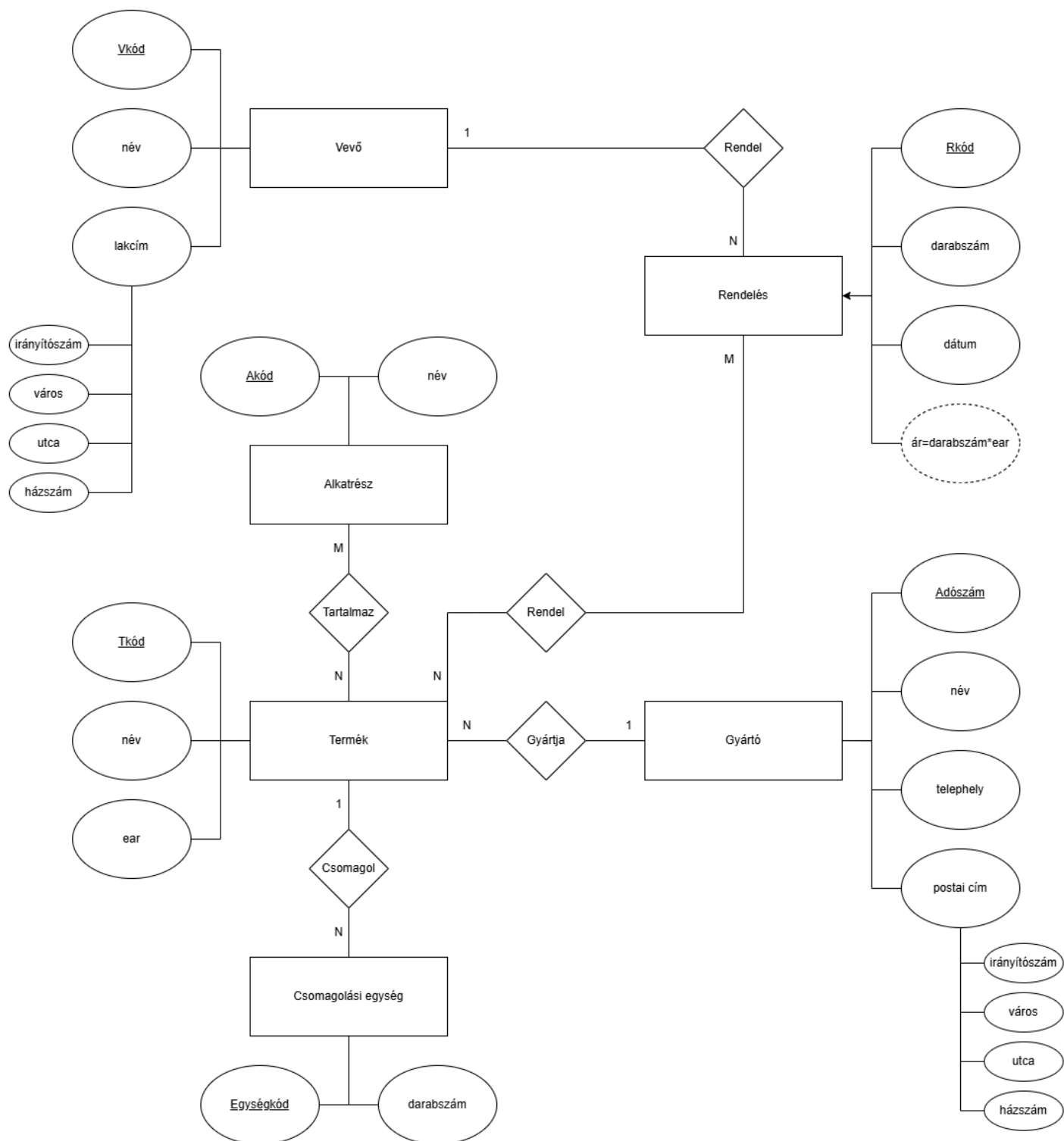
**Rendelés:** Rkód, darabszám, dátum, ár (darabszám\*ear)

**Vevő:** Vkód, név, lakcím (irányítószám, város, utca, házszám)

Majd összekapcsoljuk őket a alábbi módon:

**Rendelés M—N Termék**

**Vevő 1—N Rendelés**



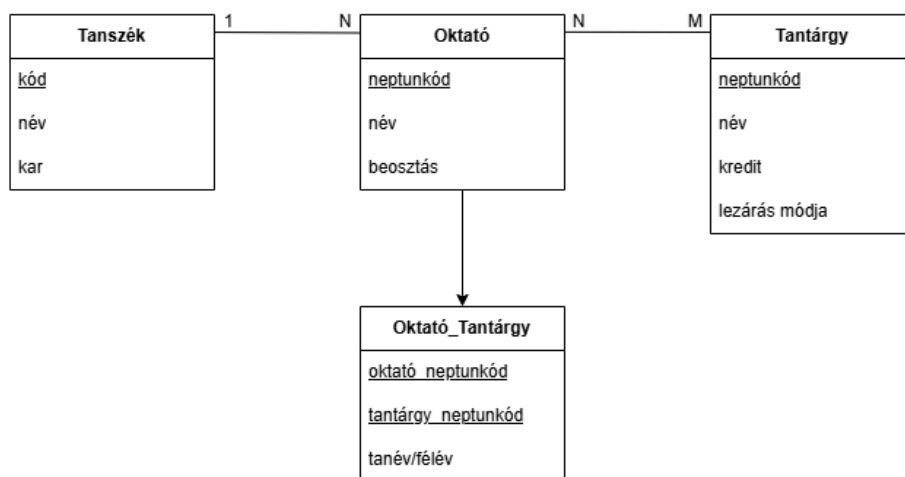
## 4. Feladat

Készítsen az ER modell, ahol relációs séma megadásával adottak a következő egyedek, hozzá tartozó tulajdonságok és kapcsolatok és tulajdonságok. A relációs modellbe (struktúra ábra) adja meg az adattípust, integritási feltételt is. Kapcsolja össze a relációkat.

**a.)** Tanszék: kód, név, kar Oktató: neptunkód, név, beosztás Tantárgy: neptunkód, név, kredit, lezárás módja  
Kapcsolatok: tanév/félév Tanszék – Oktató: 1:N Oktató – Tantárgy: N:M

**Mentés:** neptunkod\_4a.draw.io

### a.) Megoldás



**b.)** Konvertálja elkészített ER modellt relációs modellre! A relációs modellbe (struktúra ábra) adja meg az adattípust, integritási feltételeket is. Kapcsolja össze a relációkat.

**Mentés:** neptunkod\_4b.draw.io



## b.) Megoldás

Tanszék		
Attribútum	Adattípus	Integritási feltétel
kód	VARCHAR	PRIMARY KEY
név	VARCHAR	NOT NULL
kar	VARCHAR	NOT NULL

Oktató		
Attribútum	Adattípus	Integritási feltétel
neptunkód	VARCHAR	PRIMARY KEY
név	VARCHAR	NOT NULL
beosztás	VARCHAR	NOT NULL
tanszék_kód	VARCHAR	FOREIGN KEY REFERENCES Tanszék(kód)

Tantárgy		
Attribútum	Adattípus	Integritási feltétel
neptunkód	VARCHAR	PRIMARY KEY
név	VARCHAR	NOT NULL
kredit	INT	CHECK (kredit > 0)
lezárás_módja	VARCHAR	NOT NULL

Oktató_Tantárgy		
Attribútum	Adattípus	Integritási feltétel
oktató_neptunkód	VARCHAR	FOREIGN KEY REFERENCES Oktató(neptunkód)
tantárgy_neptunkód	VARCHAR	FOREIGN KEY REFERENCES Tantárgy(neptunkód)
tanév/félév	VARCHAR	PRIMARY KEY

## 5. Feladat

Konvertálja a 3. feladatban elkészített ER modellt - relációs modellre. A relációs modellbe (struktúra ábra) adja meg az adattípust, integritási feltételeket is. Kapcsolja össze a relációkat.

**Mentés:** neptunkod\_5.draw.io

## Megoldás

