

Adatbázis rendszerek I.

1. Gyakorlat

2025. 03. 11.

Készítette:

Hercegkuti Zsolt

Szak: PTI

ECCOX9

Sárospatak, 2025

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: Neptunkod1.c

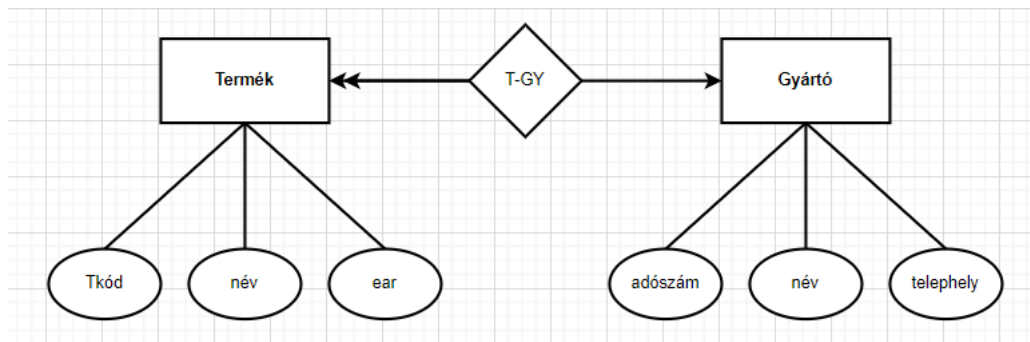
Megvalósítás:

```
1 #include <stdio.h> 35
2 #include <locale.h> 36
3 #include <stdlib.h> 37
4 #include <string.h> 38
5 39
6 #define MAX_LINE 50 40
7 41
8 void nagybeture(const char *line) 42
9 { 43
10     while (*line) 44
11     { 45
12         putchar(toupper(*line)); 46
13         line++; 47
14     } 48
15 } 49
16 50
17 int main() 51
18 { 52
19     char filename[MAX_LINE]; 53
20     char line[MAX_LINE]; 54
21     FILE *file; 55
22 56
23     printf("Adja meg a fájl nevét: "); 57
24     fgets(filename, MAX_LINE, stdin); 58
25     filename[strlen(filename)] = 0; 59
26 60
27     strcat(filename, ".txt"); 61
28 62
29     file = fopen(filename, "w"); 63
30     if (file == NULL) 64
31     { 65
32         perror("Hiba a fájl megnyitásakor"); 66
33         return 1; 67
34     } 68
35
36     printf("Add meg a neved és a neptunkódodat. A befejezéshez írd be: #\n");
37
38     while (1)
39     {
40         fgets(line, MAX_LINE, stdin);
41         if (line[0] == '#' && line[1] == '\n')
42             break;
43         fputs(line, file);
44     }
45
46     fclose(file);
47
48     file = fopen(filename, "r");
49     if (file == NULL)
50     {
51         perror("Hiba a fájl megnyitásakor");
52         return 1;
53     }
54
55     printf("\nA fájl tartalma:\n");
56     while (fgets(line, MAX_LINE, file) != NULL)
57     {
58         nagybeture(line);
59     }
60
61     fclose(file);
62     return 0;
63 }
64
```

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

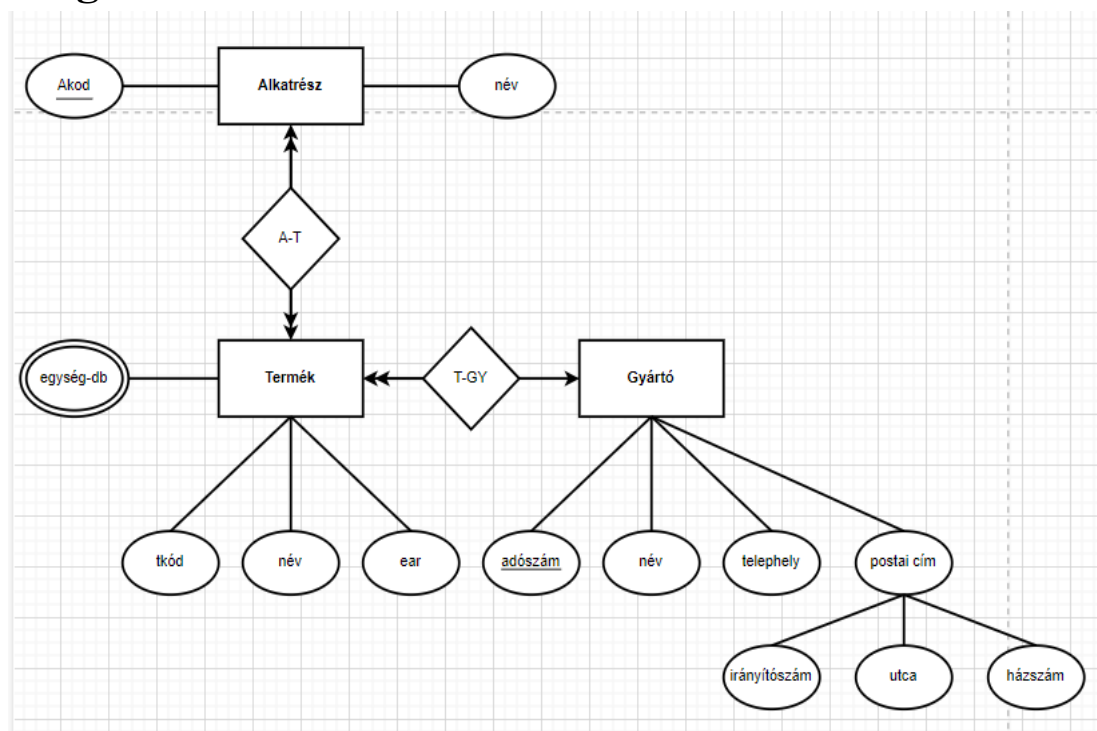
Megvalósítás:



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.

Megvalósítás:



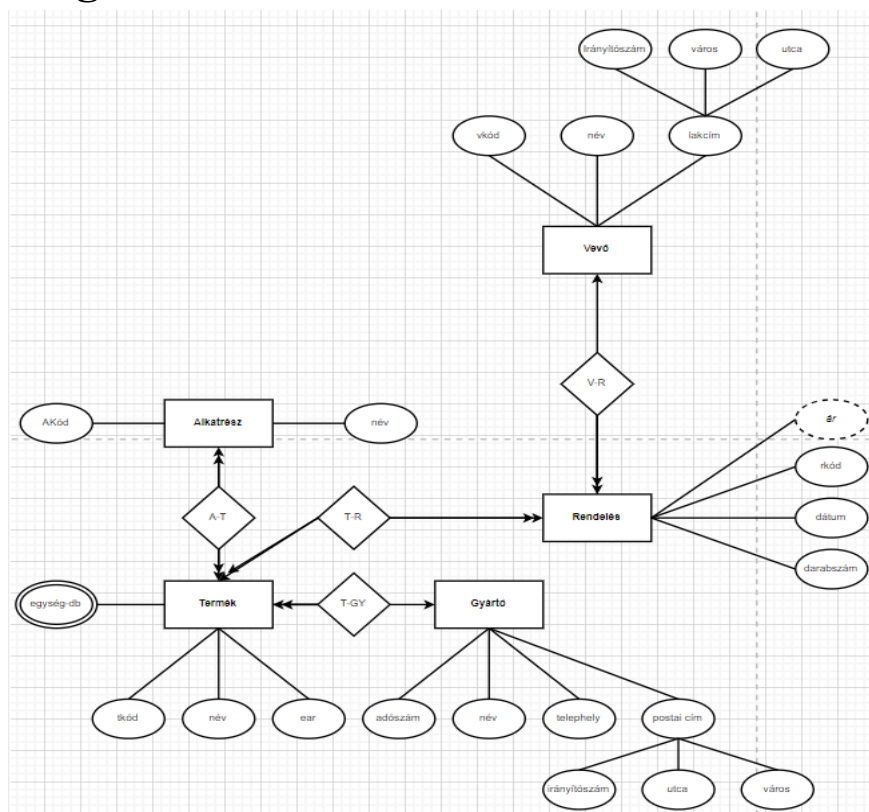
3. feladat - Az 2b. feladatnál létrehozott **Termék-Gyártó** modellt bővítse a következőképpen:

A Termék-hez a Rendelés-ek (egyed) nyilvántartása, melynél egy termékből többen is rendelhetnek és fordítva is.

A Rendelés-nél a Rkód, Dátum, Darabszám, és Ár, amely darabszám * ár formával számolódik.

A Vevő a Rendeléssel van kapcsolatba, úgy, hogy egy Vevő többször is Rendelhet. A vevő tulajdonságai Vkód, név, lakcím összetett tulajdonság.

Megvalósítás:



4. feladat - Készítsen az ER modell, ahol relációs séma megadásával adottak a következő egyedek, hozzá tartozó tulajdonságok és kapcsolatok és tulajdonságok. A relációs modellbe (struktúraábra) 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

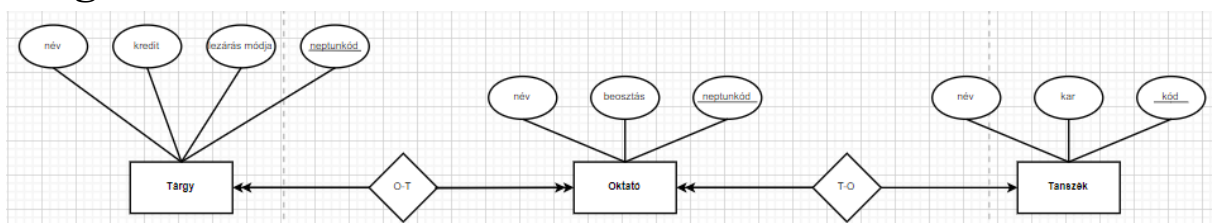
Kapcsolatok:

tanév/félév

Tanszék – Oktató: **1:N**

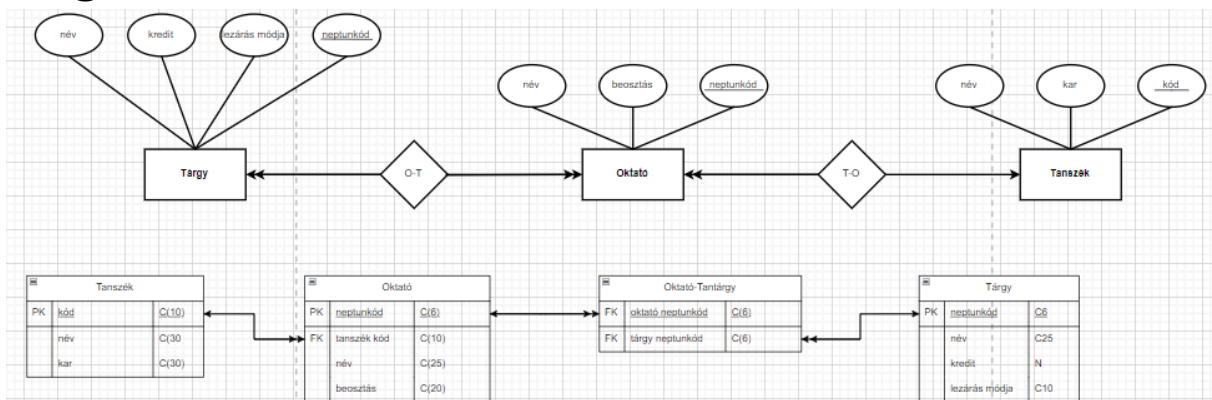
Oktató – Tantárgy: **N:M**

Megvalósítás:



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

Megvalósítás:



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.

Megvalósítás:

