

ADATBÁZIS-KEZELÉS

1. Alapfogalmak	1
1.1. Adat	1
1.2. Információ	1
1.3. Egyed, Tulajdonság, Kapcsolat	2
1.4. Adatmodellek	2
1.5. Adatbázis (DATABASE, DB)	3
2. A relációs adatmodell	4
2.1. A reláció meghatározása	4
2.2. A reláció tulajdonságai	4
2.3. A kulcs fogalma	5
2.4. Funkcionális függőségek	6
2.5. Normálformák	6
3. Tervezés	7
3.1. Információs rendszerek tervezése	7
3.2. Adatbázis-tervezés	8
3.3. Egyed – kapcsolat (E/K) adatmodell (Entity – Relationship model)	9
3.3.1. Az egyedhalmazok közötti kapcsolatok	9
3.3.2. Az egyed-kapcsolat modell jelölésrendszere	10
3.4. Példa: Áruházi adatbázis tervezése	11
4. Ajánlott irodalom	12

1. Alapfogalmak

1.1. Adat

Adat (DATA): lehet *bármí*, ami körülöttünk van, ami számunkra *jelent valamit*, valamilyen tulajdonsággal rendelkezik, mástól megkülönböztethető és a *számítógépen* valahogyan hosszú időre *tárolni szeretnénk*, azért, hogy később többször is vagy rendszeresen valamilyen formában visszakaphassuk, kombinálva a többi hasonlóan tárolt adatokkal.

1.2. Információ

Információ: valamilyen *új* ismeret, amit a tárolt adatokból kaphatunk. Nem azonos az *adattal*, hanem az a jelentés, amit mi tulajdonítunk az adatnak, amit az adat a számunkra hordoz.

Az **Információs rendszer** lehetőséget nyújt nagy mennyiségű adatok számítógépes tárolására a megfelelő hardware eszközök használatával, abból a célból, hogy a különböző felhasználók a hardware valamint a software eszközök (programcsomagok) segítségével különböző információt kaphassanak a tárolt adatokból.

Az információs rendszer az adatokat tárolja a hardware eszközökön, ahonnan a software segítségével a felhasználók információt kaphatnak.

1.3. Egyed, Tulajdonság, Kapcsolat

Egyed (entity, entitás): Az a valami, dolog, amit ismeretekkel akarunk leírni; valami ami van és megkülönböztethető. Az egyedek a valóság azon elemei, melyek számunkra valamilyen lényeges információt hordoznak, egymástól megkülönböztethetőek. Az egyed egy konkrét értékét **egyed előfordulásnak**, az előfordulások összességét pedig **egyedhalmaznak** nevezzük. Például az egyetem egyed esetén egyed előfordulás lehet az ELTE; másfelől ha az egyed alatt az iskolatípusokat értjük, akkor az egyetem, mint előfordulás szerepel. Másik példa: egyed legyen a jármű, ekkor egyed előfordulás a gépkocsi, a villamos, a repülő. Ha az alkalmazás úgy kívánja, hogy a gépkocsi legyen az egyed, akkor pl. a Kis Pista kocsija egy egyed előfordulás lesz.

Tulajdonság (attribútum): A tulajdonság az, amivel az egyedet leírjuk, ami alapján az egyedhalmaz egyedei megkülönböztethetőek a többi egyedtől. A tulajdonság egy konkrét értéke a **tulajdonság előfordulása**. A tulajdonság előfordulások összességét **tulajdonsághalmaznak** nevezzük. Például a szín, mint tulajdonság esetén a fehér egy előfordulás. Ha a tulajdonsággal a jármű fajtáját kívánjuk leírni, akkor az előfordulás lehet pl. vízi, légi, szárazföldi jármű. A típus értéke, mint a gépkocsit jellemző tulajdonság, lehet OPEL, BMW, stb., a gépkocsit jellemző másik tulajdonság lehet a gépkocsi rendszáma.

Kapcsolat (relationship): Az egyedek vagy tulajdonságaik közötti viszony. A kapcsolatokat megkülönböztethetjük annak megfelelően, hogy az egyedhalmazok közötti viszonyt vizsgáljuk, vagy az egyes egyedek tulajdonsághalmazai közötti viszonyt vizsgáljuk. Az egyedhalmazok közötti kapcsolat, a táblák (relációk) közötti kapcsolatban fog megjelenni. Az egyedhalmaz tulajdonsághalmazai közötti kapcsolatokat pedig a relációs modellnél vizsgáljuk, amikor meghatározzuk a funkcionális függőséget.

1.4. Adatmodellek

Az **adatmodell** a valóság objektumait (egyedeit), ezek tulajdonságait és a köztük lévő kapcsolatokat ábrázolja.

Az adatmodell jellemzői:

- nem konkrét értékekkel, hanem az adatok típusaival, összefüggéseivel foglalkozik
- a valóságot tükrözi
- a valóság mozgásait kell követnie

Az adatmodellt két 'dolog' határozza meg:

a jelölésrendszer, mely leírja az adatokat,

a műveletek halmaza, mely lehetővé teszi az adatok kezelését.

Az adatmodellek négy különböző fajtája ismert:

- hierarchikus
- hálós
- relációs
- egyed-kapcsolat

<i>modell</i>	<i>jelölés</i>	<i>műveletek</i>
hierarchikus modell	faszerkezet (1:N)	adatkezelő nyelv
hálós modell	háló (M:N)	adatkezelő nyelv
relációs modell	tábla	relációs algebra, SQL
egyed-kapcsolat	diagram	nincs

1.5. Adatbázis (DATABASE, DB)

Adatbázis alatt egymással valamilyen szempontból kapcsolatban lévő adatok jól szervezett halmazát értjük, mely számítógépen tartósan, hosszabb ideig tárolt és egyidejűleg több felhasználó párhuzamosan használhatja.

Az adatbázis három absztrakciós szintjét emeljük ki, ez a három szint ugyanannak az adatbázisnak három különböző vetülete:

Fizikai vagy belső szint: Meghatározza, hogy hogyan helyezkednek el az adatok fizikailag a számítógép tároló eszközén.

Fogalmi vagy konceptuális szint: A fogalmi adatbázis, az adatbázis teljes leírása egy adatmodell segítségével.

Külső, felhasználói, view (nézet) vagy alséma szint: Az adatbázis-kezelő rendszer képes minden felhasználó részére az adatbázisnak csak azt a részét megmutatni, melyre szüksége van, ez a külső vagy felhasználói szint. Ez a szint tehát egy részét adja a teljes adatbázisnak, a fogalmi adatbázisnak.

Séma és előfordulás.

Adatbázis séma: az adatbázis szerkezete, amit tervezéskor, egyszeri alkalommal hozunk létre, ritkán változtatjuk. Az adatbázis séma megadja, hogy az adatbázisban milyen egyedhalmazok vannak, ezek milyen tulajdonsághalmazokkal és kapcsolatokkal rendelkeznek. A szerkezet leírásához különböző adatmodellek állnak a rendelkezésünkre.

Adatbázis előfordulás alatt az adatbázis pillanatnyi tartalmát értjük. Általában a tervezés befejezésekor, az adatbázis használatba vételekor történik az adatok elsődleges feltöltése az adatbázisba. A mindennapi használat során az adatbázis előfordulás, a tartalom az, ami állandóan változik.

Adatfüggetlenség.

Fizikai függetlenség: egy jól szervezett adatbázisban a fizikai séma megváltoztatható, anélkül, hogy változtatni kellene a fogalmi sémán, vagy hogy újra kellene definiálni a nézeteket, a külső szintet. Ezt, a fizikai és fogalmi szint közötti függetlenséget nevezzük fizikai adatfüggetlenségnek.

Logikai függetlenség: A fogalmi és a külső szint közötti viszonyt vizsgálva beszélhetünk logikai adatfüggetlenségről. Az adatbázis használata közben szükségessé válhat a fogalmi adatbázis módosítása, például új objektumok bevezetése (új táblák létrehozása), régi objektumok új információval való kibővítése (meglévő táblákhoz új oszlop hozzáadása), feleslegessé vált objektumok megszüntetése (nem használt táblák törlése). A fogalmi sémán számos változtatás hajtható végre anélkül, hogy ezek a létező felhasználói nézeteket érintenék. Néhány, a fogalmi sémán végrehajtott változtatás igényelheti az érintett felhasználói szintek módosításait. Az alkalmazói programoknak pedig szintén az esetek többségében minden változtatás nélkül futniuk kell. Egy olyan változtatás van, ami mindenképpen a felhasználói séma átírását igényli, ha olyan információt töröltünk a fogalmi sémából, amit valamelyik alséma használt. Ebben az esetben szükségessé válhat egyes alkalmazói programok módosítása is.

2. A relációs adatmodell

2.1. A reláció meghatározása

Reláció: adott n darab halmaz, ezeket jelöljük $D_1, \dots, D_i, \dots, D_n$ -nel. Reláción e halmazok direkt szorzatának a részhalmazát értjük és R -rel jelöljük. Az R relációt tehát felírhatjuk a következő formában: $R = \{(a_1, \dots, a_i, \dots, a_n), a_i \in D_i, i=1, \dots, n\}$. A D_i halmazok a reláció értelmezési tartományai, a halmazok számát (n) a *reláció fokának* nevezzük. A *reláció számosságát* $|R|$ -el jelöljük és megadja a relációban lévő n - esek, a reláció elemeinek számát. A reláció megadására a következő jelölés szolgál: $R(D_1, \dots, D_n)$.

Az R relációt ábrázolhatjuk egy táblával, melynek n oszlopa és m sora van.

A reláció és a tábla szinonim fogalom, amennyiben a tábla rendelkezik azokkal a tulajdonságokkal, amelyek a relációt is jellemzik.

2.2. A tábla (reláció) tulajdonságai

1. Minden táblának (relációnak) egyedi neve van. pl. DIAK, DOLGOZO
2. A tábla oszlopainak (attribútumainak) neve van, és ezek a nevek egy táblán belül egyediek.
3. Az oszlopok száma (a reláció foka) az adott táblában állandó.
4. Az oszlopok sorrendje tetszőleges; azonosításuk a nevük alapján és nem a helyük alapján történik.
5. Az egyes oszlopok csak meghatározott értékeket vehetnek fel egy adott értéktartományból (D_i halmazból), ezt az értéktartományt **domain**-nek nevezzük. Minden sor minden oszlopában **egy és** csakis **egy elemi érték** szerepelhet, és egy értéknek szerepelnie is **kell**. (A gyakorlatban megengedett, hogy egy attribútum valamelyik előfordulásán ne vegyen fel értéket a domain-ből, ekkor ez az előfordulás definiálatlan lesz és **NULL értéknek** nevezzük.)
6. A **sorok** sorrendje **tetszőleges**, azonosításuk az értékük, tartalmuk alapján és nem a sorszámukkal történik. A sorok száma változhat, adott pillanatban a sorok száma megadja a reláció számosságát.
7. A táblában **minden sor különböző**. Nem létezhet két teljesen megegyező sor a táblában, mint ahogy nem létezik két azonos elem a relációban, mint halmazban sem.
8. A 7. tulajdonságból következik, hogy minden táblában (relációban) létezik az oszlopoknak egy olyan halmaza, mely a tábla (reláció) bármely sorát egyértelműen meghatározza. Ezt az oszlop kombinációt hívjuk a tábla (reláció) **kulcsának**.

A továbbiakban a relációt és a táblázatot, mint ekvivalens fogalmakat használjuk. Az adatbázis ebben a modellben relációk, azaz táblák összességét jelenti.

A relációs modell esetén a táblák írják le az egyedeket, a táblák oszlopai egy adott egyedtípus tulajdonságai vagy attribútumai, sorai az egyedhalmaz konkrét előfordulásai.

A relációs adatbázis esetén relációséma alatt a reláció nevét és attribútumainak halmazát értjük, jelölése a reláció nevével és utána zárójelben az attribútumok felsorolásával történik. A *relációs adatbázis (fogalmi) sémája* egy vagy több relációsémát tartalmaz.

Az adott reláció sorainak halmazát reláció-előfordulásnak nevezzük.

A relációs adatbázishoz tartozó relációk sorainak pillanatnyi halmazát a *relációs adatbázis előfordulásának* nevezzük.

2.3. A kulcs fogalma

Kulcs: Azt az attribútumot vagy attribútum halmazt, melynek értékei egyértelműen azonosítják a relációt, a reláció kulcsának nevezzük.

Ez a definíció séma (szerkezet) szintű, ami azt jelenti, hogy független a tábla aktuális tartalmától.

Kulcsok fajtái:

Egyszerű kulcs: a kulcs egyetlen attribútumból áll.

Összetett kulcs: a kulcsot kettő vagy több oszlop kombinációja alkotja, előfordulhat az is, hogy az összes oszlop szerepel a kulcsban.

Minimális kulcs: ha összetett kulcs esetén bármely attribútumot elhagyjuk a kulcsból, és az így megmaradt oszlopok kombinációja már nem rendelkezik kulcs tulajdonsággal, akkor az összetett kulcsot *minimálisnak* nevezzük. Az egyszerű kulcs mindig minimális.

Kulcsjelöltek: egy relációban több különböző oszlop vagy oszlopkombináció létezhet, amely elegendő tesz a minimális kulcs definíciójának, ezeket a lehetséges kulcsokat *kandidate kulcsoknak* vagy kulcsjelölteknek nevezzük.

Elsődleges kulcs (primary key): az a kulcs, melyet kiválasztunk a kulcsjelöltek közül, és kulcsként használunk. A ki nem választott kulcsjelölteket *alternatív kulcsnak* nevezzük. Az elsődleges kulcsnak nem lehet NULL az értéke.

Idegen kulcs (foreign key): olyan attribútum vagy attribútum halmaz egy adott relációban, amelyik egy másik relációban elsődleges kulcsként szerepel. Az idegen kulcsot tartalmazó relációt *hivatkozó relációnak*, a másikat, melyben ez a kulcs elsődleges, *hivatkozott relációnak* nevezzük.

Referencial vagy hivatkozási integritási szabályok:

Általános: Az adatbázisban nem lehet olyan idegen kulcs, melynek értéke nem egyezik meg a hivatkozott relációban, valamelyik elsődleges kulcs értékével.

Ennek a feltételnek az adatbázist érintő minden változtatás után érvényben kell maradnia.

Lehetséges változtatások: új adatok bevitele, meglévő adatok törlése, meglévő adatok módosítása.

2.4. Funkcionális függőségek

Definíciók:

Funkcionális függőség: adott az R reláció, azt mondjuk, hogy Y értelmezési tartománya funkcionálisan függ az X értelmezési tartományától ($X \twoheadrightarrow Y$) akkor és csak akkor, ha X minden értéke egyértelműen meghatározza Y-t. Ez a meghatározás nem csak az aktuális előfordulásokra, hanem mindig érvényes.

Funkcionális teljes függőség: adott az R reláció és az értelmezési tartományok egy halmaza X. Y értelmezési tartomány funkcionálisan teljesen függ X-től, ha Y funkcionálisan függ X-től, de nem függ funkcionálisan X egyetlen valódi részalmazától sem.

Tranzitív függőség: adott az R reláció. Z értelmezési tartomány tranzitívan függ X értelmezési tartománytól, ha Z funkcionálisan függ X-től ($X \twoheadrightarrow Z$), és Y-től ($Y \twoheadrightarrow Z$), és Y függ X- értelmezési tartománytól ($X \twoheadrightarrow Y$).

Az elsődleges kulcstól minden tartomány funkcionálisan függ, ha a kulcs egyben egyszerű is, akkor ez a függőség teljes.

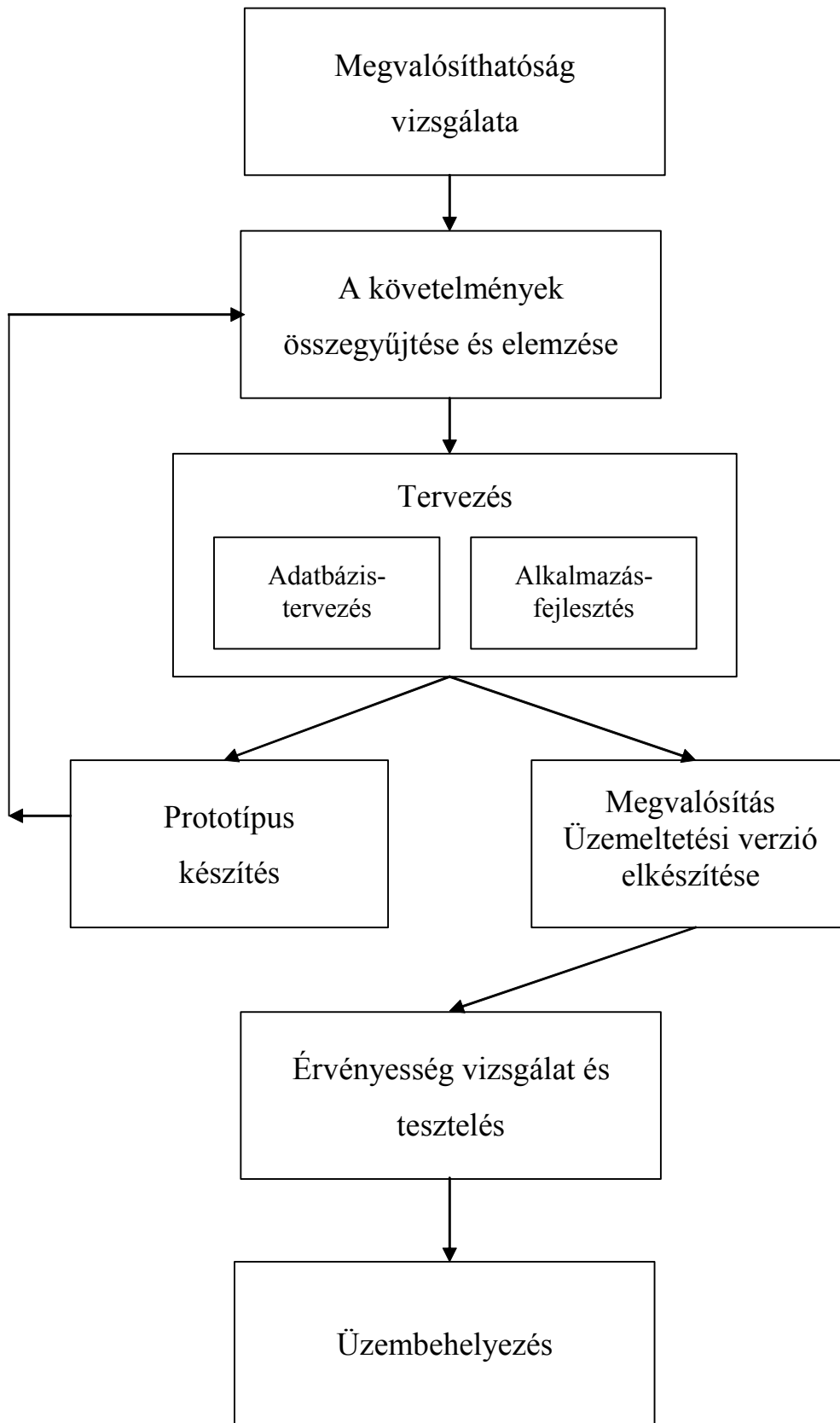
2.5. Normálformák

- 1. normálforma (1NF):** az R reláció első normálformában van, ha a relációban szereplő minden érték elemi, azaz minden attribútum minden sorban csak egy értéket vesz fel az értelmezési tartományából.
- 2. normálforma (2NF):** az R reláció második normálformában van akkor és csak akkor, ha 1NF-ben van és minden olyan attribútuma, mely nem része az elsődleges kulcsnak, funkcionálisan teljesen függ az elsődleges kulcstól.
- 3. normálforma (3NF):** az R reláció harmadik normálformában van akkor és csak akkor, ha 2NF-ben van és minden olyan attribútuma, mely nem része az elsődleges kulcsnak, funkcionálisan teljesen függ az elsődleges kulcstól **és csak attól**.

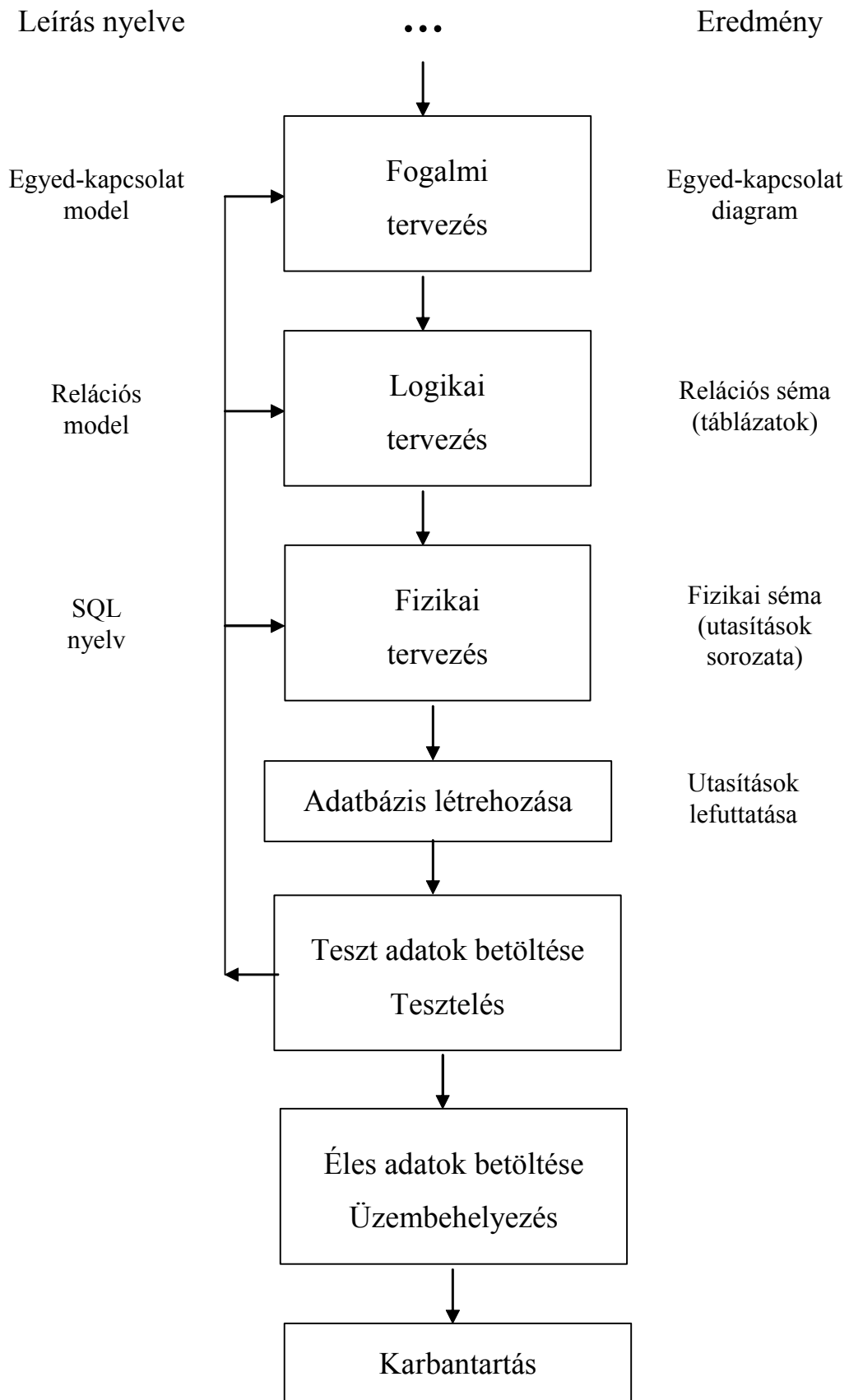
Másik definíció: Az R reláció harmadik normálformában (3NF) van akkor és csak akkor, ha 2NF-ben van és nem tartalmaz tranzitív függőségeket.

3. Tervezés

3.1. Információs rendszerek tervezése



3.2. Adatbázis-tervezés



3.3. Egyed – kapcsolat (E/K) adatmodell (Entity – Relationship model)

3.3.1. Az egyedhalmazok közötti kapcsolatok

Az egyed-kapcsolat modellben a kapcsolatok két vagy több egyedhalmazt kapcsolnak össze, mindig kétirányúak és mindkét irányban névvel írjuk le őket. Ezeket a kapcsolatokat többféle szempontból vizsgálhatjuk.

- I. Ha modellezni akarjuk a valós világot, akkor gyakran szükséges, aszerint osztályozni a kapcsolatokat, hogy a kapcsolatban álló egyedhalmazok egyes egyedeinek hány egyed felel meg a másik egyedhalmazból. A lehetséges **kapcsolattípusok**:
 - **nincs kapcsolat** az egyedhalmazok között.
 - **egy-egy (1:1) kapcsolat**. Az egyik egyedhalmaz minden egyes elemének egy és csak egy elem felel meg a másik halmazból, és viszont.
 - **egy-sok (1:N) vagy a másik irányból sok-egy (N:1) kapcsolat**. Az egyik halmaz elemeinek több elem is megfeleltethető a másik halmazból, a másik halmaz minden eleméhez viszont csak egy elem tartozik az első halmazból.
 - **sok-sok (M:N) kapcsolat**. Mindkét egyedhalmaz bármely eleméhez több elem tartozhat a másik egyedhalmazból.
- II. Az egyedhalmazok közötti kapcsolatban résztvevő egyedek számát vizsgáljuk, azt nézzük, hogy minimum hány egyede az **A** halmaznak tartozhat egy egyedéhez a **B** halmaznak. Értéke “nulla” vagy “egy” lehet. A “nulla” azt fejezi ki, hogy nem kötelező, hogy **A** minden egyedéhez tartozzon legalább egy egyed a **B**-ből, azaz a kapcsolat **tetszőleges** (optional). Az “egy” azt jelenti, hogy **A** minden egyedéhez tartozik legalább egy egyed **B**-ből, azaz a kapcsolat **kötelező** (mandatory).
- III. A kapcsolatok csoportosítása aszerint történik, hogy hány egyedhalmaz vesz részt a kapcsolatban:
 - Egy egyedhalmaz: az egyedhalmaz saját magával áll kapcsolatban, rekurzív kapcsolat.
 - Két egyedhalmaz közötti kapcsolat, ez a leggyakoribb.
 - Kettőnél több egyedhalmaz áll egymással kapcsolatban.
- IV. Speciális kapcsolat az ún. **isa** kapcsolat (az angol „is a” szókapcsolatból származik az elnevezés). Azt mondjuk, hogy **A** egyedhalmaz **isa** kapcsolatban van **B** egyedhalmazzal (**A isa B**), ha az **A** egyedhalmaz részhalmaza **B** -nek, vagy másképpen **B** kiterjesztése **A** -nak. Ebben az esetben **A** örökli **B** halmaz valamennyi tulajdonságát (attribútumát), de ezenkívül **A** rendelkezhet még egyéb attribútumokkal is, melyek nem jellemzik **B** azon elemeit, amelyek nem elemei **A** -nak is. Az **A** kulcsattribútumai, azok az attribútumok, amelyek az **a** előfordulást egyértelműen azonosítják a **B** -ben, vagyis amelyek a **B**-ben is kulcsattribútumok.

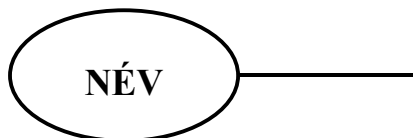
3.3.2. Az egyed-kapcsolat modell jelölésrendszere

Az egyed-kapcsolat modell grafikus megjelenítését *egyed-kapcsolat diagramnak* nevezzük. A különböző tervező eszközök a fogalmi séma létrehozásakor, az egyed-kapcsolat diagram megrajzolásánál különböző jelölési rendszerekkel dolgozhatnak, ezen jelölések egyike az ún. *CHEN jelölési rendszer*, melynek egy egyszerűsített változatát fogjuk a továbbiakban használni:

- Az egyed jelölése téglalappal történik, belsejében az egyedhalmaz nevével:



- Az egyed tulajdonságait (attribútumait) ellipszis jelzi, amit egy vonallal kötünk az egyedhez. A tulajdonság nevét az ellipszisbe írjuk, a kulcs attribútumokat aláhúzással ábrázoljuk:

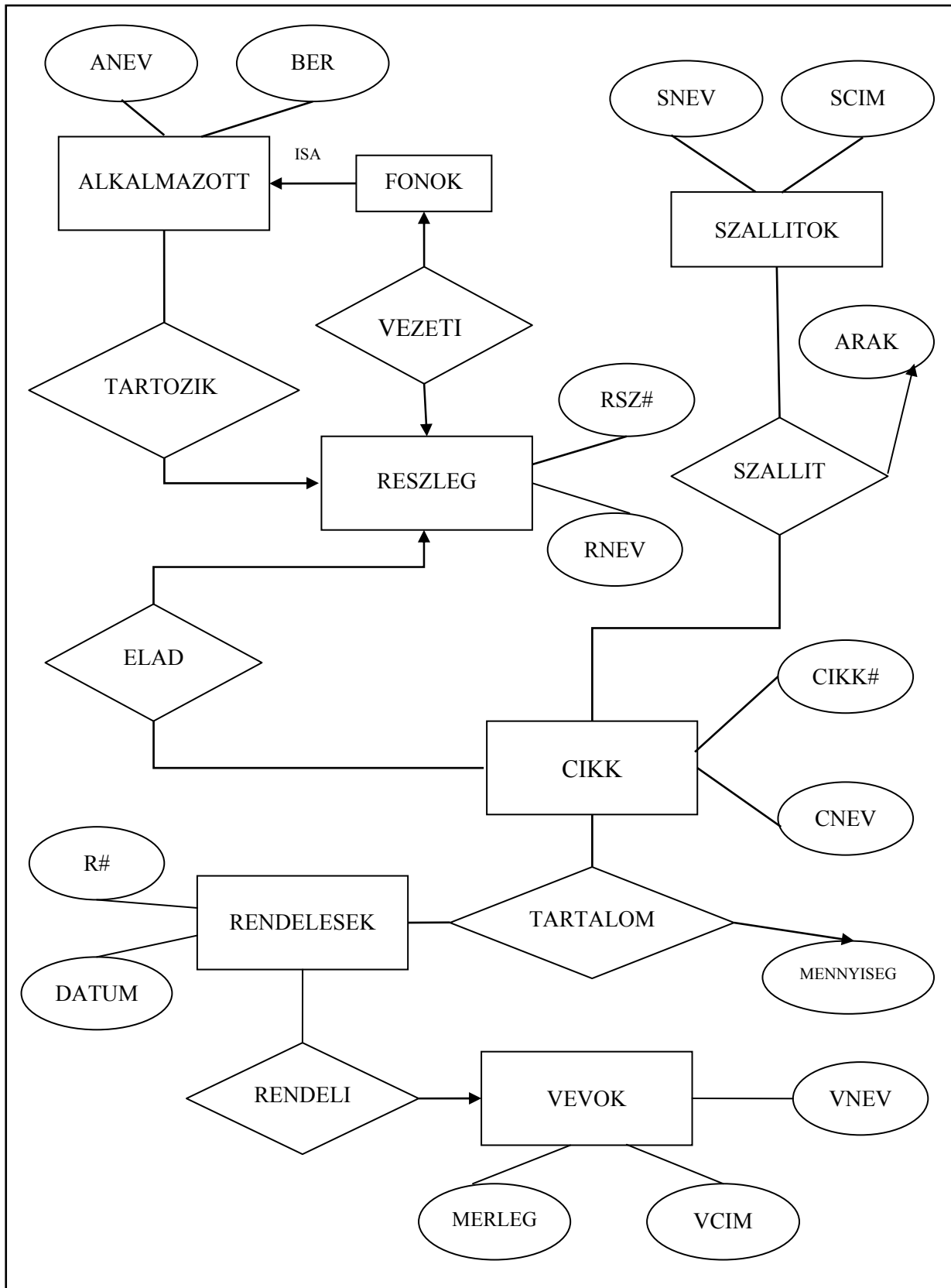


- A kapcsolatot rombuszsal jelöljük, belsejében a kapcsolat nevével, és ebben az esetben mindkét irányban ugyanez a név határozza meg a kapcsolatot:



- A kapcsolat az egyedhalmazokat irányított vagy irányítatlan éllel köti össze, annak megfelelően, hogy a két egyed között „egy” vagy „sok” jellegű a kapcsolat. A nyíl mindig az „egy” irányába mutat, a „sok” kapcsolatot összekötő él irányítatlan. Az általunk használt egyszerűsített jelölési rendszernél nem jelöljük, hogy a kapcsolat opcionális vagy kötelező. Ha **A** és **B** egyedhalmaz között *isa* kapcsolat van (**A isa B**), akkor csak a **B** felé vezető él kap nyílat.
- Előfordulhat, hogy a kapcsolatban egy egyedhalmaz csak egyetlen attribútummal rendelkezik, ilyenkor a halmaz neve lehet az attribútum neve is, és tulajdonképpen ezt nem is egyedhalmazként fogjuk jelölni, hanem attribútumként, amely közvetlenül a kapcsolathoz lesz kötve egy éllel, az ilyen tulajdonság tulajdonképpen magát a kapcsolatot jellemzi, a kapcsolat attribútumaként fogjuk kezelni.

3.4. Példa: Áruházi adatbázis tervezése.



4. Ajánlott irodalom

1. **Demetrovics, Denev, Pavlov: A számítástudomány matematikai alapjai, Tankönyvkiadó, Budapest**
2. **Halassy Béla: Az adatbázistervezés alapjai és titkai, IDG Magyarországi Lapkiadó Kft., 1994**
3. **Stolnicki Gyula: SQL kézikönyv, ComputerBooks, 2001**
4. **Jeffrey D. Ullman, Jennifer Widom: Adatbázisrendszerek Alapvetés Panem Kft, 1998.**
5. **Hector Garcia – Molina, Jeffrey D. Ullman, Jennifer Widom: Adatbázisrendszerek megvalósítása, Panem Kft, 2001.**
6. **Békéssy András, Demetrovics János: Előadások adatbázis szerkezetekből, Egyetemi jegyzet, Budapest, 1999. ELTE**
7. **Martin Gruber: Mastering SQL; SYBEX, 2000.**
8. **Martin Gruber: SQL A-Z, SYBEX KISKAPU 2003.**