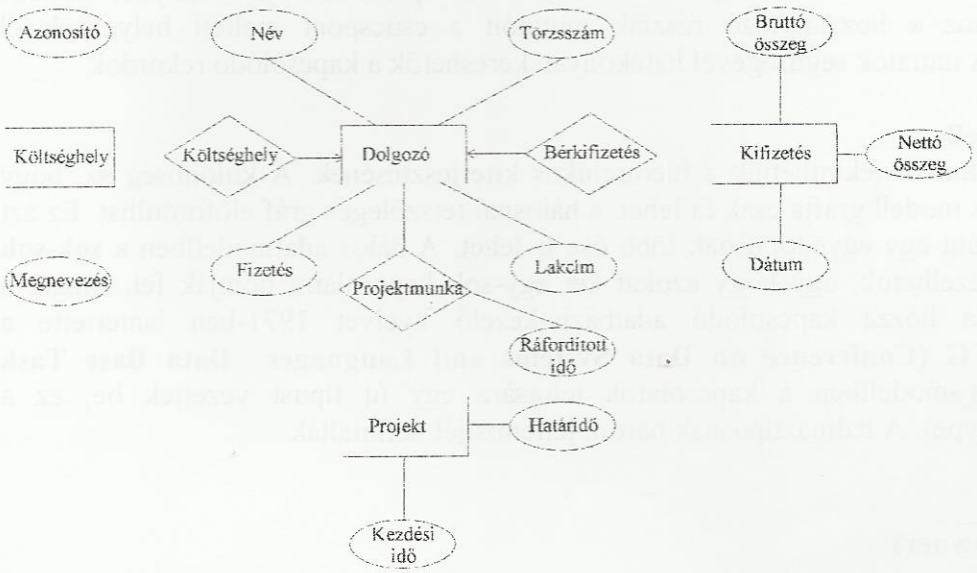
A **kapcsolatok** esetében a **rombusz** jelet használjuk. Például a Dolgozó és a Kifizetés egyedek közötti kapcsolat a következő módon reprezentálható:

5. Példa

A már említett dolgozói nyilvántartás egyed-kapcsolat diagrammja a következőképpen nézhet ki:



Az egyed-kapcsolat modellben lehetőség van több egyed közötti kapcsolat megadására is. Ez ugyanúgy történik, mint a két egyed közötti kapcsolat esetén, csak ilyenkor a kapcsolat jelét mindegyik egyeddel összekötjük. A gyakorlatban háromnál több egyed között csak nagyon ritkán alakítanak ki kapcsolatot.

Adatmodellek típusai

Az előző részben említett adatmodell elemekből különböző struktúrák igénybevételével alakítjuk ki az adatmodellt. Az adatbázis-kezelés fejlődése során három fontos adatmodellt alakítottak ki. Ezek a következők:

- **Hierarchikus modell**
- **Hálós modell**
- **Relációs modell**

A **hierarchikus és a hálós adatmodell** manapság már csak történeti okból lényeges. Ezért ezekről is csak rövid összefoglalást adunk. Ma a legaktuálisabb modellnek a relációs modell számít, ezért ennek jegyzetünkben külön fejezetet szánunk.

A **hierarchikus adatmodell:**

Hierarchikus. Azért kapta ezt az elnevezést, mert az alapelve az, hogy az egyedeket a köztük lévő kapcsolat alapján hierarchiába rendezi. A hierarchikus modell leginkább egy-egy és egy-sok jellegű kapcsolatok megvalósítására használható. A kapcsolat alapján a két egyedtípus között hierarchiát definiálunk. Fontos, hogy a hierarchiában alul lévő egyedtípusnak csak egyetlen őse lehet. A hierarchiában felül lévő egyedtípushoz több, a hierarchiában lentebb lévő egyedtípus kapcsolódhat. Az első hierarchikus ABKR az IBM által 1968-ban kifejlesztett ***Information Management System (IMS)*** volt. Mára a relációs modell már teljesen kiszorította a hierarchikust.

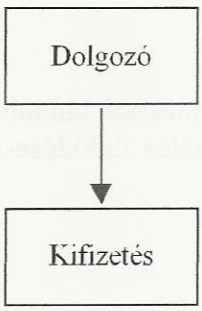
A hierarchiák alkalmazását az adatbázis-kezelésben az motiválta, hogy a fastruktúrák jól ábrázolhatók szekvenciálisan. Közismert a fáknak az a reprezentációja, amelynél minden egyes csúcsponthoz a hozzátartozó részfák mutatóit a csúcspont mellett helyezzük el szekvenciálisan. A mutatók segítségével hatékonyan kereshetők a kapcsolódó rekordok.

A hálós adatmodell

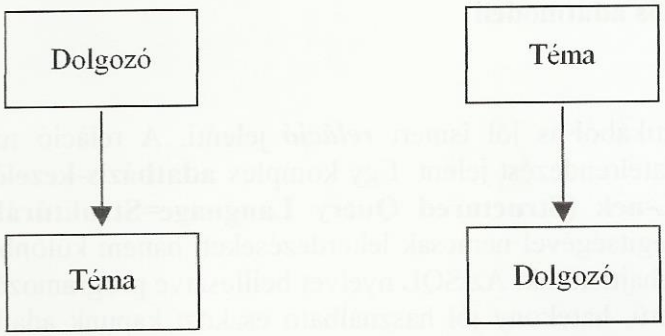
A ***hálós adatmodellt*** tekinthetjük a hierarchikus kiterjesztésének. A különbség az, hogy míg a hierarchikus modell gráfja csak fa lehet, a hálósnál tetszőleges gráf előfordulhat. Ez azt jelenti, hogy például egy egyedtípusnak több őse is lehet. A hálós adatmodellben a **sok-sok kapcsolatok** is kezelhetők, úgy hogy azokat két egy-sok kapcsolatra bontják fel. Magát a modellt, illetve a hozzá kapcsolódó adatbázis-kezelő nyelvet 1971-ben ismertette a **CODASYL DBTG (Conference on Data Systems and Languages Data Base Task Group).** A hálós modellben a kapcsolatok leírására egy új típust vezettek be, ez a ***halmaztípus*** (set type). A halmaztípusnak három jellemzőjét definiálták:

- **Név**
- **Tulajdonos (owner)**
- **Tag (member)**

A tulajdonos és a tag egyedtípusok, amelyek egyértékű és többértékű attribútumokat is tartalmazhatnak. A halmaztípus tulajdonképpen az egy-sok kapcsolat megvalósítására való. Megadása a szokásos grafikus formában történhet, azaz a két egyedtípust egy nyíllal kapcsoljuk össze. Például a **Dolgozó-Kifizetés kapcsolat** a következőképpen néz ki:



A **Dolgozó és a Téma egyedtípusok közötti sok-sok kapcsolatot** a következőképpen bonthatjuk fel egy-sok kapcsolatokra:

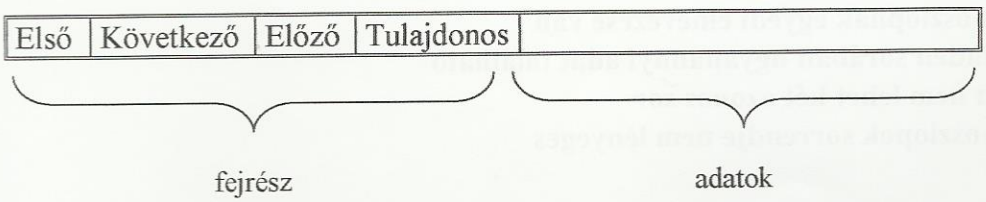


Mivel a hálós modellben nincs olyan megkötés, hogy valamely egyedtípusnak feltétlenül egy másik felett kell állnia a hierarchiában, ezért a fenti felbontás nem sérti az adatmodell alapelvét. A halmaztípus megvalósítása ugyancsak jól ismert adatszerkezettel történhet.

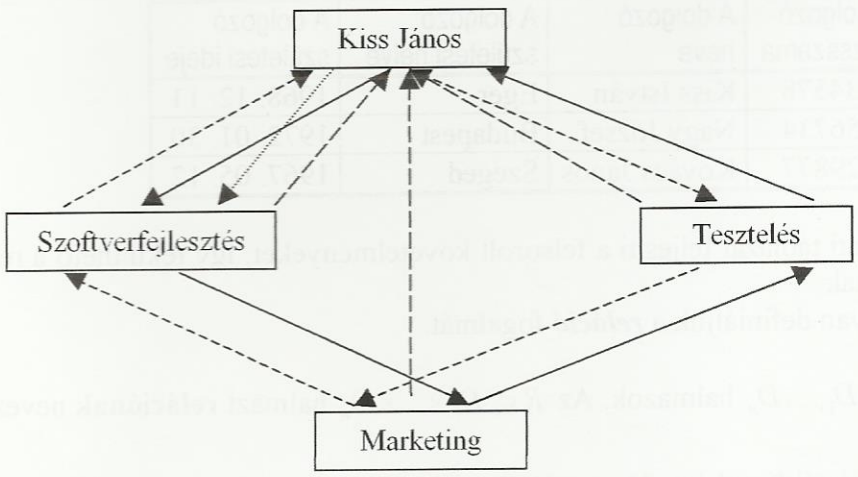
Minden egyes rekord előfordulásnál az adatok mellett egy fejrészt is képez az ABKR. Ebben a fejrészben történik a kapcsolatok megadása. A fejrészben különböző mutatók lehetnek. A leggyakrabban használt mutatófajták az alábbiak:

- **Első (first) – tulajdonostól mutat az első tagra**
- **Következő (next) – tagtól mutat a következő tagra**
- **Előző (prior) – tagtól mutat az előző tagra**
- **Tulajdonos (owner) – tagtól mutat a tulajdonosra**

A következő ábra egy rekord szerkezetét mutatja be a hálós modellben:



Végül pedig a rekordszerkezetek láncolásának ábráját láthatjuk:



Relációs adatmodell

Ennek alapját a matematikából is jól ismert ***reláció*** jelenti. A reláció tulajdonképpen **kétdimenziós táblázatos** adatelrendezést jelent. Egy komplex **adatbázis-kezelő nyelv** került kifejlesztésre, amelyet **SQL-nek (Structured Query Language=Struktúrált Lekérdező Nyelv)** neveztek el. Ennek segítségével nemcsak lekérdezéseket, hanem különböző adatbázis kezelési műveleteket is végrehajthatunk. Az SQL nyelvet beillesztve programozási nyelvekbe, fejlesztő környezetbe egyszerű, hatékony jól használható eszközt kapunk adatbázis-kezelési feladatok megoldására.

A relációs modellnek jól meghatározott elmélete van, amelynek alapjait **E. F. CODD** dolgozta ki még a 60-es évek végén, illetve a 70-es évek elején. Az elmélet bázisát a **halmazelméletből** veszi, magukat az adatokat tartalmazó táblákat halmazként definiálja, és így az adatokon végzett műveleteket is **halmazműveleteknek** felelteti meg.

A relációs adatmodell alapjai

A táblázatokra vonatkozóan különböző alapkövetelményeket szokás megkövetelni. Ezek a következők:

- **Minden egyes táblázat egyértelmű azonosítóval bír**
- **A sorok és oszlopok metszetében található adatok egyértékűek, vagy más szóval atomiak (azaz adatcsoportok, tömbök nem megengedettek). Ezeket szokásos elemi adatmezőnek is nevezni.**
- **Az oszlopokban található adatok azonos jellegűek, és egy előre definiált halmazból származnak**
- **Minden egyes oszlopnak egyedi elnevezése van**
- **A táblázat minden sorában ugyanannyi adat található**
- **A táblázatban nem lehet két azonos sor**
- **A sorok és az oszlopok sorrendje nem lényeges**

A következő táblázat egy lehetséges példát mutat, amely a **Dolgozó** nevű táblázat egy részletét ábrázolja:

A dolgozó törzsszáma	A dolgozó neve	A dolgozó születési helye	A dolgozó születési ideje
T234578	Kiss István	Eger	1968. 12. 11.
T456734	Nagy József	Budapest	1972. 01. 30.
T429877	Kovács János	Szeged	1967. 05. 12.

Látható, hogy a fenti táblázat teljesíti a felsorolt követelményeket, így tekinthető a relációs modell egy táblázatának.

Nézzük most, hogyan definiáljuk a ***reláció*** fogalmát.

Definíció: Legyenek $D_1,...,D_n$ halmazok. Az $R \subseteq D_1 \times ... \times D_n$ halmazt **relációnak** nevezzük.

A reláció felfogható táblázatként. Ha a relációnak m eleme van és ezeket $(d_1^j,...,d_n^j)$-nel jelöljük, ahol $j=1,...,m$, akkor a táblázatunk a következőképpen néz ki:

D_1	...	D_n
d_1^1	...	d_n^1
⋮		⋮
d_1^m	...	d_n^m

A táblázat sorait ***előfordulásoknak*** nevezzük.

Az előfordulások **rekordoknak** felelnek meg.

A reláció ***számosságát*** az összes előfordulás száma adja meg.

A reláció oszlopait ***attribútumoknak*** nevezzük. Minden egyes attribútumhoz tartozik egy ***tartomány.*** Ez az attribútumhoz tartozó D_i halmaz, amely azt mondja meg, hogy az adott oszlop milyen értékeket tartalmazhat. Szokás ezért a D_i halmazokat ***attribútumhalmaznak*** is nevezni. A reláció nevét és a reláció attribútumainak halmazát együtt szokás ***relációsémának*** nevezni.

Például a **Dolgozó reláció sémája** a következőképpen adható meg:

Dolgozó(A dolgozó törzsszáma, A dolgozó neve, A dolgozó születési helye, A dolgozó születési ideje)

Annak ellenére, hogy az attribútumok halmazt alkotnak (vagyis elemeinek sorrendje nem lényeges), amikor a relációsémát megadjuk, akkor a lista felsorolásával egy sorrendet is rögzítünk. Ilyen esetben az ABKR általában ezt a sorrendet szokta alkalmazni a táblázat megjelenítésekor is. Egy adatbázis több relációból áll.

Egy adatbázis relációsémáinak összességét ***relációs adatbázissémának*** nevezzük.

A reláció attribútumainak száma határozza meg a reláció ***fokát***. A relációknak két típusát definiálhatjuk, amelyek adatbázis-kezelési szempontból lényegesek.

Az ***alap reláció*** olyan reláció, amely az adatbázisban a többi relációtól függetlenül létezik. A ***származtatott reláció,*** vagy ***nézet*** más relációkból való konstruálással keletkezik. Például az adatbázisban tárolt **Dolgozó relációból** származtathatunk egy új relációt **Dolgozónő** névvel, amely reláció csak a nő dolgozókat fogja tartalmazni.

Ahogy a fentiekből kitűnik, a relációs modell egyik fontos alaptulajdonsága, hogy a táblázat sorai és oszlopai metszetében található adatok elemiek. Az attribútum halmaz elemeinek eleve **egyszerű típusúaknak** kell lenniük, hogy ezt a tulajdonságot biztosítani lehessen. De könnyen előfordulhat, hogy egy előfordulásnál valamely attribútumnak több különböző értéke van. Az egyetlen lehetőség, hogy a duplázás megszüntetésére az adott előfordulás összes többi értékét megsokszorozzuk, pontosan annyiszor, amennyi különböző érték tartozik a többértékű attribútumhoz. Így néhány új előfordulást konstruálunk, amelyek az eredetitől csak egyetlen attribútum értékeiben különböznek. Ez persze azt jelenti, hogy az adatbázisunkban megnöveljük a redundanciát. Ennek ellenére a modell **egyszerűsége, elméleti megalapozottsága,** valamint **a relációs adatbázisok egyszerű feldolgozhatósága** miatt ma a legelterjedtebb a gyakorlatban. A fenti redundancia minimálisra csökkentésére külön elméleti eljárást is kifejlesztettek.

Kapcsolatok a relációs modellben

A kapcsolatokat maguk a táblázatok tartalmazzák. Az alábbiakban ennek a megvalósítási technikáit mutatjuk be. Háromféle kapcsolatról beszéltünk:

Az **egy-egy, az egy-sok és a sok-sok** típusúakról. A relációs modellben mindegyiket különbözőképpen lehet megvalósítani.

Egy-egy kapcsolat. Ilyenkor csupán az szükséges, hogy a kapcsolatban részt vevő két egyed közül az egyiket reprezentáló tulajdonságok közül kiválasszunk **egy kulcsot.** Ezután a másik egyedhez tartozó táblázatban egy olyan attribútumot kell definiálnunk, amelynek lehetséges értékei az előzőleg kiválasztott kulcsmező értékeit vehetik fel.

Egy-sok kapcsolat esetében teljesen hasonlóan járhatunk el. Ekkor ugyanis ha annak az egyednek a szempontjából vizsgáljuk a kapcsolatot, amelynek előfordulásaihoz egy másik egyed előfordulás tartozik, pontosan ugyanazt a helyzetet kapjuk, mint az első esetben. Ebben az esetben tehát csak arra kell ügyelnünk, hogy a fenti technika egy irányban működik.

Példa

Nézzük most meg, hogyan nézhetnek ki a **Dolgozó** és a **Kifizetés** egyedek táblázatai, amennyiben a köztük lévő kapcsolatot is reprezentáljuk:

Dolgozó

A dolgozó törzsszáma	A dolgozó neve	A dolgozó születési helye	A dolgozó születési ideje
T234578	Kiss István	Eger	1968. 12. 11.
T456734	Nagy József	Budapest	1972. 01. 30.
T429877	Kovács János	Szeged	1967. 05. 12.

Kifizetés

A kifizetés azonosítója	A kifizetés dátuma	A kifizetett nettó összeg	A dolgozó törzsszáma
K23756	1999. 10. 01.	120000	T234578
K23757	1999. 11. 01.	89000	T234578
K23758	1999. 10. 01.	167000	T429877

A legbonyolultabb kapcsolatfajta a **sok-sok** kapcsolat. Sajnos ezt már nem lehet olyan egyszerű módon megadni, mint az egy-egy és egy-sok kapcsolatokat. Mivel mindkét egyed irányából nézve több előfordulás tartozhat egy előforduláshoz, egyetlen lehetőség, hogy az egymáshoz kapcsolódó előfordulásokat külön összepárosítjuk. Erre alkalmas lehet **egy speciális reláció**, vagy más néven **kapcsolótábla,** amely két attribútumból áll. Az egyik attribútum az egyik egyed, a másik a másik egyed egy kulcsának lehetséges értékeit veheti fel értékül. Az összetartozó előfordulásokat a hozzájuk tartozó kulcsértékeknek a kapcsolótáblában való összepárosításával adjuk meg.

Példa

Vizsgáljuk most meg a Dolgozó és a Téma relációk kapcsolatának ábrázolását a relációs modellben. Segítségül egy Dolgozótéma nevű új táblát definiálunk, amely a kapcsolat megadására való. Feltételezzük, hogy a Dolgozó tábla ugyanaz, mint az előző példánál.

Téma

A téma azonosítója	A munka kezdete	Befejezési határidő	Témavezető neve
T25	1998. 06. 01.	1999. 12. 31.	Nagy Ádám
T26	1999. 01. 01.	2000. 06. 01.	Mekk Elek
T27	1999. 03. 01.	2002. 12. 31.	Remek Jenő

Dolgozótéma

A téma azonosítója	A dolgozó törzsszáma
T25	T234578
T25	T456734
T25	T429877
T26	T456734
T26	T429877
T27	T456734

Funkcionális függések és jellemzésük

A relációs modell egyik legfontosabb fogalma a funkcionális függés. Segítségével a táblázat attribútumai között bizonyos összefüggéseket állapíthatunk meg. Vagyis **egy funkcionális függés megadja azt, ha az adott relációban valamely attribútumokon vett értékek meghatározzák más attribútumok értékeit.** Segítségével ugyanis csökkenthető a modell redundanciája, ugyanis a funkcionális függésben lévő attribútumokat külön relációkba szervezhetjük, anélkül hogy ezzel bármilyen információt elveszítenénk.

Definíció: Legyen R a $D_1,...,D_n$ halmazokon értelmezett reláció. Legyen továbbá $A,B \subset \{D_1,...,D_n\}$ két attribútumhalmaz. Azt mondjuk hogy B **funkcionálisan függ** A**-tól** az R relációban ($A \rightarrow B$), ha bármely két $r1,r2 \in R$-re $r1(D_i) = r2(D_i), \forall D_i \in A$ esetén teljesül, akkor ebből következik hogy $r1(D_j) = r2(D_j), \forall D_j \in B$-re.

Példa

Tekintsük most ismét a **Dolgozótéma relációt**, és tegyük fel, hogy ezúttal a következő módon terveztük meg

Dolgozótéma

A dolgozó törzsszáma	Téma azonosítója	Témavezető neve
T234578	T25	Nagy Ádám
T456734	T25	Nagy Ádám
T429877	T25	Nagy Ádám
T456734	T26	Mekk Elek
T429877	T26	Mekk Elek
T429877	T27	Remek Jenő

Láthatjuk, hogy ha a Téma azonosítója oszlop értékei megegyeznek, akkor a Témavezető oszlop értékei is megegyeznek. Igy a fenti definíció alapján teljesül a **Téma azonosítója -> Témavezető funkcionális függés**. Azt is könnyen felfedezhetjük, hogy a függés egyfajta redundanciát jelent, ugyanis felesleges háromszor leírni, hogy a T25 azonosítójú projekt vezetője Nagy Ádám.

Az első fontos kérdés, - ami gyakorlati szempontból is jelentős - hogy ha ismerünk bizonyos függőségeket, akkor ezekből kikövetkeztethetünk-e újabb függőségeket. Amennyiben igen, akkor a következtetési szabály alkalmazásával egy függőséghalmazból újabb függőségeket vezethetünk le.

Armstrong-axiómák. Ezek segítségével már meglévő függőségekből új függőségek származtathatók.

Az egyik legegyszerűbb következtetési szabály azt mondja ki, hogy ha az $A, B, C \subseteq R$ attribútumhalmazokra teljesülnek az $A \to B$ és $B \to C$ függőségek, akkor teljesül az $A \to C$ függőség is. Ezt a tulajdonságot nevezzük ***tranzitivitásnak.***

Nyilvánvalóan igaz az a szabály is, hogy a B halmaz minden B' részhalmazára is teljesül az $B \to B'$ függőség. Ezt nevezzük ***triviális függőségnek***, vagy ***reflexivitásnak.***

A következő két szabály szorosan kapcsolódik egymáshoz, mivel lényegében egymás ellentettjei.

Az első az ***egyesítési szabály***, amely azt mondja ki, hogy ha teljesül egy $A \to B$ és egy $A \to C$ függőség, akkor teljesül az $A \to B \cup C$ függőség is. Szavakkal kifejezve azonos bal oldalú függőségek jobb oldalán szereplő attribútumhalmazait egyesíthetjük.

Ugyanakkor a funkcionális függőség jobb oldalán szereplő attribútumhalmazra teljesül az, hogy annak minden részhalmaza is függ a baloldaltól. Vagyis ha $A \to B$ és $C \subseteq B$, akkor $A \to C$ is teljesül. Ezt nevezik ***szétvághatósági szabálynak.***

Ugyancsak fontos a ***bővítési szabály.*** Ez azt mondja ki, hogy ha egy funkcionális függés mindkét oldalát ugyanazzal az attribútumhalmazzal bővítjük, akkor a függés továbbra is megmarad a két halmaz között.

Formálisan ha $A \to B$ és C tetszőleges attribútumhalmaz, akkor $A \cup C \to B \cup C$.

Kulcsok a relációs modellben

A relációs modellben külön definíciót adtak a kulcsra. Ez tulajdonképpen megfelel a korábban már ismertetett kulcs fogalomnak, csak a relációs modellben pontos matematikai jellegű definíció is adható.

Definíció: Legyen $A = \{A_1, ..., A_n\}$ az R reláció attribútumhalmaza. A $K \subseteq A$ halmazt a **reláció kulcsának** nevezzük, ha

(1) a K attribútumain felvett értékek egyértelműen meghatározzák a reláció elemeit, azaz $K \to A$, valamint

(2) nincs K-nak olyan valódi részhalmaza amelyre ugyanez teljesül, azaz K **minimális.**

Az olyan attribútumhalmazokat,a melyekre csak az (1) tulajdonság teljesül ***szuperkulcsnak*** nevezzük.

Meg szoktuk különböztetni azt az esetet, amikor a kulcs egy attribútumból áll. Az ilyen jellegű kulcsokat ***egyszerű kulcsoknak*** nevezzük.

Ellenkező esetben ***összetett kulcsokról*** beszélünk.

Egy relációnak természetesen több kulcsa is lehet, és minden egyes attribútumról eldönthető hogy tartozik-e kulcshoz vagy sem.

Elsődleges attribútumoknak nevezzük azokat az attribútumokat, amelyek valamely kulcshoz tartoznak,

másodlagosaknak azokat, amelyekre ez nem teljesül.

A relációs modell elméletében a kulcsok között nem szokás különbséget tenni, a gyakorlatban azonban gyakran szükséges hogy meghatározzunk egy **elsődleges kulcsot**, amely főleg az ABKR számára fontos a rekordok fizikai elhelyezéséhez.

A definíció alapján a kulcshoz tartozó attribútumokon **egy értéksorozat csak a reláció egyetlen sorában fordulhat elő**. Ugyanis mivel a kulcs a teljes attribútum halmazon felvett értékeket meghatározza, ezért ha lenne két olyan sor, amelyekben a kulcshoz tartozó attribútumokon azonos értékek vannak, akkor két azonos sornak kellene lenni a relációban. Ez pedig nem lehetséges.

Az olyan attribútumokat, amelyek egy másik relációban alkotnak kulcsot ***külső kulcsnak***, vagy ***idegen kulcsnak*** szokás nevezni. A külső kulcsoknak a **kapcsolatoknál** van jelentősége. Mint láttuk, egy kapcsolatot a relációs modellben úgy lehet reprezentálni, hogy az egyik kapcsolódó táblázatban, vagy esetleg külön relációban külső kulcsot, vagy kulcsokat szerepeltetünk.

Külső kulcsok különböző formában jelenhetnek meg a relációs modellben. Előfordulhat az, hogy **egy kulcs saját táblájában szerepel külső kulcsként**. Például elképzelhető hogy a bérszámfejtő rendszerben tárolni kívánjuk hogy az egyes dolgozóknak ki a főnöke. Ebben az esetben a Dolgozó tábla a következő lehet:

Dolgozó(A dolgozó törzsszáma, A dolgozó neve, A dolgozó születési helye, A dolgozó születési ideje, A vezető törzsszáma)

A vezető törzsszáma attribútum tulajdonképpen külső kulcs, hiszen adatait a A dolgozó törzsszáma kulcs értékei közül veszi fel, azonban az a speciális helyzet áll elő, hogy a külső kulcs éppen ugyanahhoz a táblázathoz tartozó kulcs. Ez egyfajta **rekurzív kapcsolatot** jelent. Az is előfordulhat, hogy egy táblában egy külső kulcs többféle szerepben található meg.

Például ha a

Dolgozótéma(A dolgozó azonosítója, Téma azonosítója, Témavezető neve)

relációnál a **Témavezető neve** attribútum helyett a **Témavezető törzsszámát** használjuk (amennyiben kapcsolatot kívánunk létesíteni e szerint az attribútum szerint a Dolgozó táblával akkor ez mindenképpen szükséges), akkor ebben a relációban **két idegen kulcs** lesz.

Relációk normalizálása

A tervezés során olyan adatstruktúrákat alakítsunk ki, amelyek segítik a hatékony adatkezelést. **Fontos hogy egy-egy táblába csak a valóban logikailag összetartozó adatok kerüljenek, és hogy minél kevesebb ismétlődés legyen az adatok között.**

A relációs modellben külön eljárást fejlesztettek ki arra vonatkozóan, hogy az adatok megfelelő strukturálását, a redundancia csökkentését elősegítsék.

Ez a módszer a ***normalizálás.***

Bevezettek ***első, második, harmadik, Boyce-Codd, negyedik, ötödik*** **normálformát** is. Ezek közül a legnagyobb jelentősége a **harmadik és a Boyce-Codd** normálformáknak van, mivel ezek azok, amelyek a tényleges gyakorlati problémákat is lefedik, és nem csak nagyon speciális esetben fordulnak elő. Az első és második normálformák egyes nézetek szerint eléggé kezdetlegesek, jelentőségük inkább csak történeti, mivel az előbb említett két fejlettebb normálforma tartalmazza ezeket is. A negyedik és ötödik normálformák inkább elméleti jelentőségűek, így ezekkel nem foglalkozunk.

A legfontosabb műveletek a **beszúrás, a módosítás és a törlés.** Nézzük, milyen problémák merülhetnek fel ezeknél.

Adatkezelési műveletek anomáliái

Beszúrási anomáliáról beszélünk abban az esetben, amikor egy adatrekord beszúrása egy másik, hozzá logikailag nem kapcsolódó adatcsoport beszúrását kívánja meg.

Például ha a **Dolgozótéma relációt** az alábbiak szerint terveztük meg:

Dolgozótéma(A dolgozó törzsszáma, A dolgozó neve, A dolgozó születési helye, A dolgozó születési ideje, A vezető törzsszáma, Téma azonosítója, Témavezető)

Ebben az esetben egy kézenfekvő kulcs a **A dolgozó törzsszáma, Téma azonosítója** pár. A korábban látottak szerint egyedül egyik sem alkothat kulcsot, ha így tervezzük a táblát. Ebben az esetben viszont csak akkor tudjuk egy új dolgozó adatait felvinni, ha az rögtön elkezd dolgozni egy témán, és meg tudjuk adni a téma azonosítóját is. Ez azonban nem biztos, hogy mindig teljesül, ugyanis előfordulhat hogy a dolgozónak nem kell önálló témán dolgoznia. Láthatjuk, hogy a problémát az okozza, hogy egymástól logikailag különálló dolgokat - a **dolgozó és a témák adatait** - egy táblázatban helyeztük el, és egy bizonyos adatcsoport bevitele egy másik, tőle logikailag különálló adatcsoport bevitelét kívánja meg.
Ebben az esetben fennáll a **beszúrási anomália.**
Kiküszöbölésére az adatbázis tervét úgy kell átalakítani, hogy ilyen eset ne fordulhasson elő.

A következő problémát a már felvitt **adatok módosítása** okozhatja. Abban az esetben, ha egy relációban egy adat módosítása több helyen történő módosítást igényel, akkor ***módosítási anomáliáról*** beszélünk.

Tekintsük ismét a fenti példát. Tegyük fel, hogy a próbaidőt sikeresen teljesítő dolgozó hirtelen több témán történő munkára kap megbízást. A relációs modell sajátossága szerint minden egyes témára meg kell ismételnünk a dolgozó adatait. Tegyük fel, hogy ezek után a dolgozó nevet változtat, és szeretnénk végrehajtani a módosítást az adattáblában. Ekkor minden egyes olyan rekordnál módosítást kell alkalmazni, ahol az adott dolgozó adatai szerepelnek.

Annyi módosításra van szükség, ahány témán a dolgozó dolgozik vagy dolgozott. A probléma akkor merül fel, amikor 100 ezer vagy milliós nagyságú tétel van. Ekkora mennyiségű adat módosítása már egy gyors számítógép számára is időigényes lehet.

A harmadik anomália az **adatok törlésekor** fordulhat elő. Amennyiben egy adat törlésével másik, hozzá logikailag nem kapcsolódó adatcsoportot is elveszítünk, ***törlési anomáliáról*** beszélünk.