

## 28. fejezet

# Adatbázisok – adatmodellezés

Az adatbázis-kezelő rendszerek feladata, tulajdonságai. Egyed-kapcsolat modell, relációs adatmodell, az E/K diagram átalakítása relációs adatmodellbe. Adatbázisok tervezése, anomáliák, funkcionális és többértékű függőségek, implikációs probléma, attribútumhalmazok lezárása, dekompozíciók tulajdonságai, veszteségmentesség, függőségörzés ellenőrzése, Boyce-Codd normálforma, 3NF, 4NF, dekomponáló algoritmusok.

### 28.1. Az adatbázis-kezelő rendszerek

Az adatbázis-kezelő rendszerek megszületését nagy mennyiségű adathalmaz tárolásának és hatékony kereshetőségének igénye indukálta, még a lyukszalagos számítógépek korában. Ma adatbázis-kezelő rendszer alatt olyan számítógépprogramot értünk, mely megvalósítja nagy tömegű adat biztonságos tárolását, gyors lekérdezhetőségét és módosíthatóságát, tipikusan egyszerre több felhasználó számára.

Az adatbázis-kezelő rendszerek fő jellemzője az alkalmazott adatmodell. Az első, széles körben elterjedt és mindmáig leggyakrabban használt modell a *relációs modell*, ezen kívül ismertek még az *egyed-kapcsolat modell*, illetve az *objektumorientált* és a *logikai modellek*.

Az adatbázis-kezelési tevékenységeket két csoportra szokás osztani: *adatmanipulációra* (lekérdezés), illetve *definiálásra* (adatszerkezetek kialakítása, módosítása). Az adatok manipulációjára szolgáló nyelveket összefoglalóan *Data Manipulation Language*-nek (DML), míg a definíciós eszközökkel rendelkező nyelveket *Data Definition Language*-nek (DDL) szokás nevezni.

## 28.2. Adatmodellek

### 28.2.1. Egyed-kapcsolat modell

Az egyed-kapcsolat modellben az adatok szerkezetét grafikusan, *egyed-kapcsolat diagramon* ábrázoljuk. A diagram elemei:

- egyedhalmazok,
- attribútumok,
- kapcsolatok.

**Egyedhalmaz.** Az egyedhalmazok hasonló *egyedek* (absztrakt objektumok) halmazát jelölik. Jelölése *téglalappal* történik.

**Attribútum.** Az egyedhalmazokhoz tartozó, az egyedek tulajdonságait leíró objektumok. Az attribútumok atomiak, tehát nincs belső szerkezetük. Jelölése *oválissal* történik.

**Kapcsolatok.** Kapcsolat két vagy több egyedhalmazt köthet össze egymással. A kapcsolatokat általában névvel látjuk el. Jelölése *rombusszal* történik.

Az egyed-kapcsolat modell tehát egy olyan gráf, ahol a csúcsoknak egyedhalmazok, illetve attribútumok, az éleknek pedig kapcsolatok felelnek meg.

**Előfordulás.** Az egyed-kapcsolat diagram által leírt adatbázissémát aktuális adataival együtt nevezzük a modell egy előfordulásának. Tehát egy előfordulásban *egyedek egy konkrét halmaza* szerepel.

### Kapcsolatok

A kapcsolat két oldalán álló egyedek száma szerint megkülönböztetünk:

- sok-egy,
- egy-egy és
- sok-sok

kapcsolatokat. A kapcsolatok jellegét nyilakkal jelöljük, a nyilak a kapcsolatok azon ágai felé mutatnak, amelyek "egyszeresen" vesznek részt a kapcsolatban.

A kapcsolatok rendelkezhetnek szerepekkel, ezáltal egy egyedhalmaz több szerepben is részt vehet egy kapcsolatban.

Rendelhetünk továbbá a kapcsolatokhoz *attribútumok*at is. Az attribútumok értékei a kapcsolathalmazhoz rendelt táblázat egy-egy sorához tartoznak. Az attribútumok azonban kiválthatók új egyedhalmaz bevezetésével.

Sokágú kapcsolatokat *bináris* alakíthatunk új, attribútumok nélküli (vagy a kapcsolat attribútumait hordozó) egyedhalmaz bevezetésével.

Speciális kapcsolat az *öröklődés* („az-egy”) kapcsolat, mely az objektum-orientált öröklődéshez hasonló kapcsolatot jelöl egyedhalmazok között. (Az ilyen kapcsolatokat *háromszöggel* jelöljük.)

## Megszorítások

Fent már megismertük a kapcsolatok nyilazását, mely a funkcionális függőségek kifejezésére szolgál.

**Kulcsok.** Egy egyedhalmaz kulcsa attribútumainak egy olyan halmaza, amelynek értékei egyértelműen meghatározzák az egyedhalmaz példányait. A kulcsattribútumokat aláhúzással jelöljük.

Minden egyedhalmaznak rendelkeznie kell kulccsal (a gyenge egyedhalmazok esetében ez lehet más egyedhalmaz attribútuma is).

**Hivatkozási épség.** A hivatkozási épség megkövetelését egy kapcsolatban kerek nyílvéggel jelezzük.

**Részt vevő egyedek száma.** A kapcsolat résztvevőinek számára vonatkozó megszorítást a nyíl mindkét végén jelezhetjük a megszorítás megadásával. (Például  $\leq 10$ .)

## Gyenge egyedhalmazok

A gyenge egyedhalmazok olyanok, melyeknek bizonyos kulcsattribútumai nem saját attribútumok. A gyenge egyedhalmazokat, illetve a kulcsukat tartalmazó egyedhalmazok felé mutató *kiteljesítő kapcsolatokat* dupla kerettel jelöljük.

A kiteljesítő kapcsolatoktól elvárjuk, hogy *sok-egy* (vagy *egy-egy*) kapcsolatok legyenek, melyek megkövetelik a hivatkozási épséget is (tehát biztosítani kell, hogy a gyenge egyedhalmaz minden eleméhez pontosan egy kiteljesítő egyedhalmaz kötődjön a kapcsolat mentén).

### 28.2.2. Relációs modell

**Reláció.** A relációs modell alapelemei a *relációk*. Ezek kétdimenziós táblázatok, ahol a sorokat *rekord*nak is nevezzük, és egy összetartozó adategyüttest reprezentálnak, míg az oszlopok neveit *attribútum*nak hívjuk.

A relációk sorait halmazként tekintjük, azaz sorrendjük nincs. A reláció oszlopai is felcserélhetők (az oszlopnevekkel együtt).

**Értéktartományok.** A modellben elvárjuk, hogy minden sor minden komponense *atomi* legyen, és minden oszlophoz tartozzék egy *értéktartomány* (azaz egy elemi típus), mely meghatározza az oszlopban szereplő értékek típusát.

**Séma.** Egy reláció sémáját az attribútumok és azok értéktartományai definiálják. A séma tehát megadja a táblázat szerkezetét.

**Előfordulás.** Egy adott reláció sorainak halmazát előfordulásnak (vagy a séma előfordulásának) nevezzük.

### Megszorítások

**Kulcsok, kulcsmegszorítások.** Attribútumok egy halmaza *szuperkulcsot* alkot a relációra nézve, ha egy sort kulcshalmazának értékei egyértelműen meghatároznak a reláción belül. Egy attribútumhalmaztól megszorításképp elvárhatjuk, hogy kulcsot alkosson a relációban. A szuperkulcs *kulcs*, ha nincs olyan valódi részhalmaza, amely *szuperkulcs*.

**Hivatkozási épség.** A hivatkozási épség megszorítással elvárhatjuk, hogy egy adott reláció valamely attribútumának minden értéke előforduljon egy másik reláció másik attribútumának értékeként ( $\pi_A(R) \subseteq \pi_B(S)$ ).

**Típusmegszorítás.** Megszorítást tehetünk attribútumok érték-halmazának szűkítéseként.

**Megszorítások relációs algebrával.** Számos megszorítást kifejezhetünk relációs algebrai kifejezésekkel. Ezek formája, ha  $R$  tetszőleges reláció (relációs algebrai kifejezés):  $R = \emptyset$  vagy  $R \subseteq S$ .

### 28.2.3. Egyed-kapcsolat modell átírása relációs modellé

1. Minden egyedhalmazt írjunk át az attribútumaival együtt relációvá (a reláció attribútumai legyenek az egyedhalmaz attribútumai)!
2. Egy kapcsolatot helyettesítsünk olyan relációval, amelynek attribútumai a kapcsolatban álló egyedhalmazok kulcsának felelnek meg (hozzávéve a kapcsolat attribútumait)!

#### Megjegyzések

- A többirányú kapcsolatok binárisra alakítás után alakíthatók relációkká.
- Egyes kapcsolatok (sok-egy, illetve egy-egy kapcsolatok) mentén bizonyos relációk összevonhatók, így elhagyhatjuk a kapcsolatot reprezentáló relációt.
- A gyenge egyedhalmazokat kicsit – de nem sokkal – nehezebb átírni relációkká. A kiteljesítő kapcsolatokat nem írjuk át relációvá, viszont a gyenge egyedhalmazhoz tartozó relációban, illetve nem kiteljesítő kapcsolataiban az egyedhalmaz minden kulcsa szerepel attribútumként.

## 28.3. Relációs adatbázisok tervezése

### 28.3.1. Függőségek

#### Funkcionális függőség

**28.3.1. Definíció.** *A funkcionális függőség azt fejezi ki, hogy ha  $R$  két sora megegyezik az  $A_1, \dots, A_n$  attribútumokon, akkor meg kell egyezniük más  $B_1, \dots, B_m$  attribútumokon is.*

A funkcionális függőségeket jelölhetjük például  $A_1, \dots, A_n \rightarrow B_1, \dots, B_m$  alakban.

Könnyen látható, hogy a funkcionális függőség a kulcsmegszorítás kiterjesztése: ha  $A_1, \dots, A_n$  kulcs, akkor a kulcsmegszorítás szerint a két sornak az összes többi attribútumon is meg kell egyeznie – azaz a két sor azonos.

**Következés és ekvivalencia.** A két fogalom a funkcionális függőségekre természetesen definiálható.

**Szétvágás és összevonás.** Egy funkcionális függőség, melynek jobb oldalán több attribútum szerepel, *szétvágható* olyan függőségek  $m$  elemű sorozatára, melyek jobb oldalán csak egy-egy attribútum szerepel. A szétvágás „fordítottja”, azaz azonos bal oldalú függőségek *összevonása* is lehetséges.

**Triviális függőségek.** Egy funkcionális függőség triviális, ha jobb oldala a bal oldal részhalmaza.

Egy funkcionális függőség jobb oldaláról elhagyva a bal oldalon is szereplő attribútumokat, az eredetivel ekvivalens függőséghez jutunk.

**Tranzitivitás.** A „funkcionális függőségben állni” reláció (mely két attribútumhalmaz között értelmezett), tranzitív.

**Vetítés.** Egy reláció egy attribútumhalmazra vonatkozó vetületére meghatározhatók az eredeti reláció függőségi halmazából a vetületre érvényes függőségek.

**Bázis, minimális bázis.** Egy függőség-halmazra nézve egy másik, vele ekvivalens függőség-halmaz bázis. Egy függőség-halmaz minimális bázisára az alábbiak igazak:

- minden elemének jobb oldalán egyetlen attribútum áll,
- ha bármely függőséget elhagyjuk, akkor a halmaz már nem bázis,
- ha bármely függőség bal oldaláról elhagyunk egy attribútumot, akkor a halmaz már nem bázis.

Megjegyzendő, hogy minimális bázis nem tartalmaz triviális függőséget.

### Többértékű függőség

**28.3.2. Definíció.** Ha  $R$  relációra teljesül az  $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$  többértékű függőség, akkor minden olyan  $t, u$  sorpárra, melyek megegyeznek az  $A$ -kon, találunk a relációban olyan  $v$  sort, mely megegyezik:

- $t$ -vel (és  $u$ -val) az  $A$ -kon,
- $t$ -vel a  $B$ -ken és

- $u$ -val a maradék attribútumokon.

Azaz ha rögzítjük az  $A$ -k értékeit, akkor a  $B$ -k értékei függetlenek az összes többi attribútumok ( $C$ -k) értékétől.

**Tulajdonságok.** A tranzitivitás és a triviális függőség a korábbiakhoz hasonló.

Egyszerűen belátható, hogy minden funkcionális függőség egyben többértékű is ( $v := u$ ).

### 28.3.2. Normálformák

A normálformák célja, hogy kiküszöböljük segítségükkel az adatbázis használata során keletkező anomáliák lehetőségét. A normál formára hozás minden esetben olyan átalakítás lesz, melynek során az eredeti relációban tárolt adatokat pontosan reprezentálják a normálforma adatai (azaz nem veszünk információt).

#### Anomáliák

**Redundancia** Az információk felesleges ismétlődése több sorban.

**Módosítási anomália** Egy érték módosítását nem végezzük el mindenhol, ahol az érték szerepel.

**Törlési anomália** Ha az értékek halmaza üressé válik, akkor mellékhatásként más információkat is elveszíthetünk.

#### Dekompozíció

Az anomáliák megszüntetésének elfogadott módja a dekompozíció, azaz a relációk felbontása. Egy relációt úgy bontunk fel, hogy attribútumait két (nem feltétlenül diszjunkt) halmazra osztjuk, és a megfelelő attribútumhalmazokra vetítjük a reláció tartalmát.

#### Attribútumhalmaz lezártja

Egy  $A$  attribútumhalmaznak az  $S$  függőség-halmazra nézve lezártja azon  $B$  attribútumok halmaza, melyekre igaz, hogy  $S$  elemeiből következik  $A \rightarrow B$ .  
Jele:  $A^+$ .

A triviális függőségeket is megengedjük, így  $A$  elemei mind benne vannak  $A^+$ -ban. A lezárt kiszámítása  $S$  szabályainak lehető legtöbb szétvágása után az  $A$  halmazból kiindulva már könnyű.

### Veszteségmentesség, függőségek megőrzése

**Veszteségmentes összekapcsolás.** Egy séma szétvágása az  $F$  függőség-halmazra nézve veszteségmentesen összekapcsolható, ha feltéve, hogy az eredeti reláció teljesíti a függőségeket, a vágás utáni relációk természetes összekapcsolása visszaadja az eredeti relációt (nem a sémát).

**Függőségek megőrzése.** Azt mondjuk, hogy egy szétvágás függőségőrző (megőrzi az  $F$  függőség-halmazt), ha az  $F$ -ből a szétvágás elemeire vonatkozó függőségek uniójának lezárása megegyezik  $F$  lezárásával (tehát a szétvágásból és az eredeti függőség-halmazból pontosan ugyanazon függőségek vezethetők le).

### Boyce-Codd normálforma (BCNF)

**28.3.3. Definíció.** Egy reláció BCNF-ban van, ha minden nem triviális funkcionális függőségének bal oldala szuperkulcs.

**Felbontás.** Legyen kezdetben  $R$  reláció és  $S$  függőség-halmaz.

1. Ha  $R$  BCNF-ban van, akkor az eljárásnak vége.
2. Különben válasszunk egy  $X \rightarrow Y$  függőséget, ami sérti a normálforma szabályait!
3. Legyen  $R_1 = X^+$ ,  $R_2 = X \cup (R \setminus X^+)$ !
4.  $R_1$ -re és  $R_2$ -re rekurzívan alkalmazzuk az eljárást, miután meghatároztuk függőségeiket!

### 3. normálforma (3NF)

Ez a normálforma a BCNF gyengítése, mivel nem minden reláció bontható fel BCNF formára.

**28.3.4. Definíció.** Egy reláció 3NF-ban van, ha minden nem triviális függőségre igaz, hogy



- *a bal oldala a reláció superkulcsa, vagy*
- *a jobb oldal azon attribútumai, amelyek nem szerepelnek a bal oldalon, egy kulcsnak az elemei (nem feltétlenül ugyanannak a kulcsnak).*

**Felbontás.** Legyen  $R$  reláció és  $F$  a funkcionális függőségeinek halmaza.

1. Keressük meg  $F$  egy minimális bázisát, legyen ez  $G$ !
2. A  $G$ -ben szereplő minden  $X \rightarrow A$  funkcionális függőségre legyen a felbontás eleme az  $XA$  sémájú reláció!
3. Ha ez előző lépésben kapott relációk attribútumhalmazainak egyike sem superkulcs  $R$ -ben, adjunk még egy relációt az eredményhez, melynek sémája kulcs az  $R$ -ben.

#### 4. normálforma (4NF)

A 4. normálforma feladata a többértékű függőségek által okozott redundancia kiküszöbölése.

**28.3.5. Definíció.** *Egy reláció 4NF-ban van, ha minden nem triviális többértékű függőségének bal oldala superkulcs.*

**Felbontás.** Kezdetben  $R$  relációt  $S$  funkcionális és többértékű függőségek halmazát tekintjük.

1. Ha  $R$  4NF-ban van, akkor kész vagyunk.
2. Különben keressünk egy olyan függőséget, amely megsérti a 4NF szabályát!
3. Osszuk fel a relációt úgy, hogy  $R_1$  sémáját adják a függőség mindkét oldalán található attribútumok;  $R_2$  sémáját pedig álljon a függőség bal oldali attribútumaiból, továbbá azokból, amelyek nem szerepelnek a függőségben.
4. A két részrelációra vonatkozó függőségek megkeresése után alkalmazzuk rájuk ugyanezt az algoritmust!