

Adatbázisok – 3. gyakorlat

Normalizálás

2012. október 17.

Lehet-e funkcionális függés alapú redundancia az így létrejött felbontás alapján elkészített adatbázisban? Függőségőrző-e a sémafelbontás?

Minden részséma BCNF, tehát funkcionális függőség alapú redundancia nincs.

A $R_6(PMD)$ részséma a legérdekesebb: D nem szerepel a függéshalmaz bal oldalain, $M^+ = M$, $P^+ = PC$, de abból csak P esik a részsémába. Azonban $PM \rightarrow CM$ és $CM \rightarrow PD$, így $(PM)^+ = PCMD$, a részsémára vetítve PMD . Így F vetületének a fedése $PM \rightarrow D$. A gondolatmenetet érdemes jól megnézni, mert felhasználtunk egy olyan attribútumot, ami nem is ebben a részsémában van (lásd 5. feladat). A vetületekből a következő függéshalmaz áll elő:

$$F_2 = \{IT \rightarrow K, T \rightarrow OSCM, P \rightarrow C, PM \rightarrow D\}.$$

Ez vajon egyenértékű az eredeti:

$$F = \{IT \rightarrow K, T \rightarrow OSCM, CM \rightarrow PD, P \rightarrow C\} \text{-fel?}$$

Sajnos nem, az F -beli $CM \rightarrow PD$ függés nem vezethető le F_2 -ből, mert $(CM)^+(F_2) = CM$.

A felbontás tehát sajnos nem függőségőrző. Ennek következtében külön illesztés műveletek elvégzése nélkül nem tudjuk ellenőrizni, hogy a hiányzó függőséget megőrzi-e egy update. Ha tehát betesszük az adatbázisba egy új gyártó termékét (R_4 -be insert), akkor hiába van már azonos funkciójú és méretű eszköz, simán lehet, hogy az új szerszámokat egy másik polcra tesszük fel (amelyen R_5 miatt azonos funkciójú termékek vannak, de esetleg más méretekből), és a díjazás egységességét is csak polcon belül tudjuk biztosítani (R_6).

Az eredeti relációs sémát a tanult módszerrel bontsuk fel veszteségmentes és függőségőrző módon legalább 3NF részsémákba! Törekedjünk arra, hogy minél kevesebb számú részséma keletkezzen!

Van erre tanult módszer, amely éppen a 3NF-séget garantálja, a BCNF szintet már nem tudja. ehhez először el kell készíteni F minimális fedését. Először hozzuk olyan formára, hogy minden függés jobb oldalán csak egy attribútum álljon! Így az egyes függőségek „feldarabolásával” az:

$$F' = \{IT \rightarrow K, T \rightarrow O, T \rightarrow S, T \rightarrow C, T \rightarrow M, CM \rightarrow P, CM \rightarrow D, P \rightarrow C\}$$

függéshalmaz adódik.

Második lépésben azt kell megvizsgálni, hogy valamelyik függés bal oldalán van-e felesleges attribútum; az egyetlen attribútumból állók nyilván nem egyszerűsíthetők, $IT \rightarrow K$ egyszerűsítésével $I \rightarrow K$ vagy $T \rightarrow K$ keletkezne, azonban az eredeti függéshalmazból egyik se vezethető le ($I^+ = I, T^+ = OSCMPD$), így ezekkel *erősebbé* tennénk a függéshalmazt; hasonlóan zárhatóak ki $CM \rightarrow P, CM \rightarrow D$ egyszerűsítései.

Harmadik lépésben megvizsgáljuk, hogy van-e elhagyható függés. Ehhez egyenként ellenőrizzük a függőségeket, pl. a $T \rightarrow S$ függés nélkül $T^+ = TOCMPD$ adódna, amiben az S nincs benne, tehát az elhagyásukkal *gyengébbé* tennénk a függéshalmazt; végül az adódik, hogy nincs elhagyható függés. Tehát F_{min} valóban egy minimális fedése F -nek.

A következő, garantáltan függőségörző, 3NF sémafelbontást készíthetjük el F_{min} alapján:

$$\{ITK, TO, TS, TC, TM, CMP, CMD, PC\}$$

Nem kell egy kulccsal kibővítenünk, mert ez egyben veszteségmentes is, hiszen az első részséma, ITK R -ben szuperkulcs, ugyanis tartalmazza IT kulcsot.

A részsémák száma tovább csökkenthető, ha néhányat összejoinolunk közülük – ez a veszteségmentességet és a függőségörzőséget nyilván nem befolyásolja, csak a 3NF-re kell vigyázni. PC kapásból elhagyható, mert CMP tartalmazza. TO, TS, TC, TM összevonható $TOSCM$ -mé, ami 3NF (korábban beláttuk). CMP, CMD összevonható $CMPD$ -vé, ami szintén 3NF (szintén volt már). Végül visszakaptuk a korábban már levezetett $R_1(ITK), R_3(CMPD), R_4(TOSCM)$ felbontást. (Már megállapítottuk, hogy sajnos nem BCNF).