



Adatbázisok

Boyce-Codd Normálforma

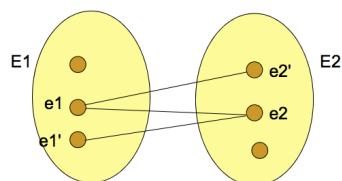
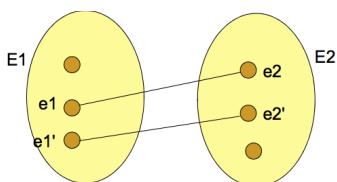
- Def: Az R reláció BCNF-ben van akkor és csak akkor, ha minden olyan esetben, ha R-ben érvényes egy $X \rightarrow Y$ nemtriviális függőség, vagyis Y nem része X-nek, akkor az X halmaz R szuperkulcsa.
 - Megnézzük az attribútum halmazok lezártjait egyesével, amik kulcsot alkotnak azokat kulcsként tüntetjük föl, és azok a függőségek fogják sérteni a relációt amik nem kulcsok.
- Triviális függőség
 - Olyan $X \rightarrow Y$ függőségek, amelyekre: $Y \subseteq X$.
- Szuperkulcs
 - Azokat az attribútumhalmazokat, amelyek tartalmazznak kulcsot, szuperkulcsoknak nevezzük (kulcsnál bővebb halmaz). Minden kulcs egyben szuperkulcs is.

Feladat:

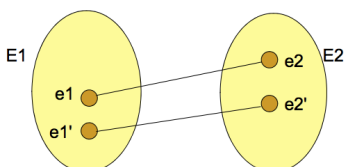
$A \rightarrow D$ $(A)^+ = \{A, D\} = (AD)^+$ $(AB)^+ = \{A, B, D\}$ Sértik: $A \rightarrow D$, $DE \rightarrow B$
 $AC \rightarrow E$ $(AC)^+ = \{A, D, C, E, B\} \Rightarrow$ Kulcs
 $DE \rightarrow B$ $(AE)^+ = \{A, E, D, B\}$

E/K diagram

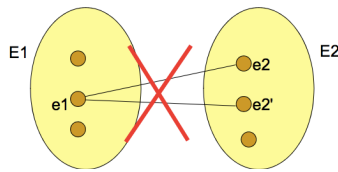
- E/K diagram: séma-szinten grafikusán ábrázoljuk. Egyedhalmazok: téglalap
Tulajdonságok: ovális A kapcsolatok jele: rombusz
 - Egy-Egy
 - Sok-Sok



- “is-a” (speciális egy-egy kapcsolatot, például “PC is a Computer”, azaz minden E1-hez pontosan egy E2 szerepel)



- Sok-Egy (Egy E1-hez legfeljebb egy E2, de fordítva igaz lenne)



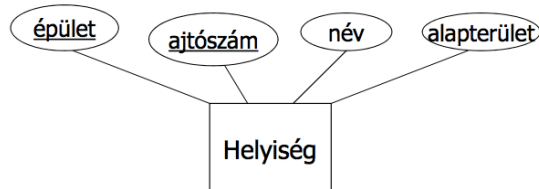
- 3 féle átírás:

- **E/R stílusban:** Egy reláció minden alosztályra, de az általános osztályból csak a kulcsokat vesszük hozzá a saját attribútumokhoz.
- **Objektumorientált stílusban:** Egy reláció minden alosztályra, felsorolva az összes tulajdonságot, beleértve az örökölteket is.
- **Nullértékek használatával:** Egyetlen reláció az öröklődésben résztvevő összes osztályra. Ha egy egyed nem rendelkezik egy alosztály speciális tulajdonságával, akkor ezt az attribútumot NULL értékkel töltjük majd ki.

- Egyszerű kulcs: név a Sörök elsődleges kulcsa:



- Összetett kulcs: épület, ajtószám két-attribútumos elsődleges kulcsa a Helyiség-nek:



- E/K átírása:

- Egyedhalmazok:
 - KÖNYV (könyvszám, szerző, cím)
 - OLVASÓ (olvasószám, név, lakcím)
- Kapcsolatok:
 - KÖLCSÖN (könyvszám, olvasószám, kivétel, visszahozás)



Idegen kulcs / Kulcs

- Def: Az SQL-ben egy reláció azon attribútumát vagy attribútumait idegen kulcsnak nevezzük, amelyek egy másik reláció bizonyos attribútumaira hivatkoznak.
 - A másik reláció azon attribútumait, amelyekre hivatkozunk elsődleges kulcsként vagy UNIQUE-ként kell deklarálni abban a relációban.
 - Az idegen kulcs értékeinek, amelyek előfordulnak az első relációban, elő kell fordulniuk a hivatkozott attribútumokban is a másik reláció valamelyik sorában.
- Megadása:
 - <attribútumnév> <típus> REFERENCES relációnév(attribútum lista)
 - FOREIGN KEY (attribútum lista) REFERENCES relációnév (attribútum lista)
- **Védekezés idegen kulcs megszorítás megsértése ellen:** Előfordulhat olyan eset, hogy az adatok frissítése sértené az idegen kulcs megszorítást. Ezek megkerüléséhez, a tábla deklarálásakor meg kell adni, hogy a kulcs ellenőrzése késleltethető (DEFERRABLE) vagy nem (NOT DEFERRABLE). Alap esetben NOT DEFERRABLE. Ha egy megszorítást DEFERRABLE-ként deklarálunk, lehetőség van arra, hogy a megszorítás ellenőrzését késleltesse a tranzakció végéig. A DEFERRABLE szót követheti INITIALLY DEFERRED (tranzakció végéig késleltetve van az ellenőrzés) vagy INITIALLY IMMEDIATE (az ellenőrzés minden egyes utasítás után azonnal megtörténik) kiegészítés.
- **Kulcs:** Azt mondjuk, hogy az egy vagy több attribútumból álló $\{A_1, A_2, \dots, A_n\}$ halmaz az R kulcsa, ha:
 - Ezek az attribútumok funkcionálisan meghatározzák a reláció minden más attribútumát, nem lehet R-ben két olyan különböző sor, amely mindegyik A_1, A_2, \dots, A_n -nen megegyezne (Egyedi a reláció soraira nézve)
 - Nincs olyan valódi részhalmaza $\{A_1, A_2, \dots, A_n\}$ halmaznak, amely funkcionálisan meghatározná R összes többi attribútumát, azaz a kulcsnak minimálisnak kell lennie.
- Megadása:
 - Egyszerű: <attribútumnév> <típus> PRIMARY KEY vagy <attribútumnév> <típus> UNIQUE
 - Összetett: A CREATE TABLE utasításban az attribútum deklaráció után a kiegészítő részben meg lehet adni további tábla elemeket: *PRIMARY KEY (attrnév₁, attrnév_k)*

Funkcionális függőség

- Def: Ha R két sora megegyezik az A_1, A_2, \dots, A_n attribútumokon, akkor meg kell egyezniük más attribútumok egy B_1, B_2, \dots, B_m sorozatán. Formálisan $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$ –mel jelöljük és azt mondjuk hogy: „ $A_1 A_2 \dots A_n$ funkcionálisan meghatározza $B_1 B_2 \dots B_m$ -et.

Korrelált alkérdés

- Def: Olyan, beépített alkérdés, mely többször értékelődik ki és minden egyes kiértékelés megfelel egy olyan értékadásnak, amely az alkérdésen kívüli sorváltozóból származik.

Kurzorok

- Def: Iterátorok ahhoz, hogy adatbázisok sorait tudjuk kezelni PL/SQL-ben

- Implicit:

```
FOR ciklusváltozó_név N (  
    SELECT utasítás  
)  
LOOP  
    utasítások;  
END LOOP;
```

- Explicit:

- CURSOR név [paraméterlista] [RETURN sortípus] IS select utasítás;

- Használat:

- Megnyitás:

```
OPEN kurzornév [aktuális paraméterlista];
```

- Betöltés változóba:

```
FETCH {kurzornév | kurzorváltozó név}  
INTO {rekordnév | változónév lista}
```

- Bezárás:

```
CLOSE {kurzornév | kurzorváltozó név};
```

Hivatkozási épség

- Def: Ha egy érték megjelenik valahol egy környezetben, akkor ugyanez az érték egy másik, az előzővel összefüggő környezetben is megjelenik. Relációs algebrai kifejezéssel: $\pi A(R) \subseteq \pi B(S)$

Chase teszt

- Adott $R=ABCD$ relációs séma és $F = \{A \rightarrow B, B \rightarrow C, AC \rightarrow D\}$ funkcionális függőségek halmaza. $d = \{AB, BC, CD\}$ dekompozíció veszteségmentességét döntsük el Chase-teszt algoritmussal.

Tabló:

| A | B | C | D |
|----------------|----------------|----------------|----------------|
| a | b | c ₁ | d ₁ |
| a ₂ | b | c | d ₂ |
| a ₃ | b ₃ | c | d |

Az első két sor megegyezik B-n, ezért $B \rightarrow C$ miatt $c_1 = c$

| A | B | C | D |
|----------------|----------------|---|----------------|
| a | b | c | c ₁ |
| a ₂ | b | c | d ₂ |
| a ₃ | b ₃ | c | d |

Most a sorok megegyeznek C-n, de nincs olyan szabály, ami C-ből következne.

Nincs olyan sor, ahol ne lennének indexelt tagok, ezért a felbontás nem veszteségmentes.

Dekompozíció

- Def: R felbontása azt jelenti, hogy R attribútumait szétosztjuk úgy, hogy két új reláció sémáját alakítjuk ki belőlük. Legyen adott R reláció $\{A_1, A_2, \dots, A_n\}$ sémával. R-t felbonthatjuk $S(B_1, B_2, \dots, B_m)$ és $T(C_1, C_2, \dots, C_k)$ relációkra úgy, hogy:
 - $\{A_1, A_2, \dots, A_n\} = \{B_1, B_2, \dots, B_m\} \cup \{C_1, C_2, \dots, C_k\}$
 - $S = \pi_{B_1, B_2, \dots, B_m}(R)$
 - $T = \pi_{C_1, C_2, \dots, C_k}(R)$
- A felbontás célja, hogy egy relációt többel helyettesítsünk úgy, hogy ezzel megszüntessük az anomáliákat.

3as Norma

- Def: Valahányszor létezik R-ben egy $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$ nem triviális függőség, akkor vagy az $\{A_1, A_2, \dots, A_n\}$ halmaz az R superkulcsa, vagy azokra az attribútumokra B_1, B_2, \dots, B_m közül, amelyek nincsenek az A-k között, teljesül, hogy egy kulcsnak az elemei
- $X \rightarrow B$ eleme F^+ (Vagy X az R superkulcsa, vagy B kulcsnak az eleme)
- Ha mindegyik atribútum kulcs, akkor tutira 2NF a forma

SQL → Reláció algebra

- Legyenek adva az $R(A,B,C)$ és $S(C,D)$ sémájú relációk. Írja át az alábbi SQL lekérdezéseket kiterjesztett relációs algebrai kifejezésre:

a)

```
SELECT A, AVG(D)
FROM R NATURAL JOIN S
WHERE B >= 2
GROUP BY A;
```

$$\pi_{A, \text{AVG}(D)} (\gamma_A (\sigma_{B \geq 2} (R \bowtie S)))$$

b)

```
SELECT A
FROM R NATURAL JOIN S
GROUP BY A
HAVING AVG(D) > 10;
```

$$\pi_A (\sigma_{\text{AVG}(D) > 10} (\gamma_A (R \bowtie S)))$$

c)

```
SELECT DISTINCT A
FROM R, S
WHERE R.C = S.C AND B = 2
ORDER BY A;
```

$$\tau_A (\delta_A (\sigma_{R.C = S.C \wedge B = 2} (R \times S)))$$

DATALOG

- Műveletek leírásának szabályai a következőkből épülnek fel:
 - egy fejnek nevezett relációs atom
 - ezt követi a \leftarrow szimbólum, amelyet gyakran „if”-nek olvasunk ki
- ezután következik a törzs, amely egy vagy több részcélnak nevezett atomból áll, amelyek lehetnek relációs vagy aritmetikai atomok. A részcélokat AND-del kötjük összes, szükség esetén használható a NOT is.

HosszúFilm(fc,é) \leftarrow Filmek(fc,é,h,m,s,p) AND h > 100

- Akkor igaz a HosszúFilm(fc,é), ha létezik a Filmek relációnak egy olyan sora, amelyre teljesülnek a következők:
 - a filmcím és év attribútumoknak megfelelő két első komponens értéke fc és é.
 - a hossz attribútumnak megfelelő harmadik komponens (h) értéke legalább 100
 - a negyedik, ötödik, hatodik komponens értéke tetszőleges.
- HA egy változó mindösszesen egyszer szerepel a szabályban (tehát nem használjuk, csak argumentumként szerepel), azt kipótolhatjuk _ jelekkel:
HosszúFilm(fc,é) \leftarrow Filmek(fc,é,h,_,_,_) AND h > 100
- Datalogban egy lekérdezés egy vagy több szabály együttese. Ha egynél több relációt tartalmaz a szabályfej, akkor ezen relációk egyike lesz a lekérdezés eredménye, míg a maradék az eredmény megfogalmazásában segít. Meg kell jelölnünk, melyik reláció adja meg a lekérdezésre a választ.
- A szabályban szereplő változók felveszik az összes lehetséges értéket, amikor a változók értékei igazzá teszik az összes részcélt, akkor megkapjuk, hogy mi a fej értéke az aktuális változókra és hozzáadjuk a kapott sort a fejben szereplő prediktátumnak megfelelő relációhoz.
- „Mely (x,y) várospárokra lehetséges egy vagy több átszállással eljutni x városból y városba?”

Datalog:

Eljut(x,y) \leftarrow Jaratok(l,x,y,i,é)
Eljut(x,y) \leftarrow Eljut(x,z) AND Eljut(z,y)

SQL (WITH):

```
WITH RECURSIVE Eljut(honnan,hova) AS
(SELECT honnan, hova FROM Jaratok)
UNION
(SELECT R1.honnan, R2.hova
FROM Eljut R1, Eljut R2
WHERE R1.hova = R2.Honnan)
SELECT * FROM Eljut
//(SELECT hova FROM Eljut WHERE honnan ='DEN'
```