

Adatbázisok elmélete 10. előadás

Csima Judit
Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi Tsz.
I. B. 136/b
csima@cs.bme.hu
2003. Március 18.

ADATBÁZISOK ELMÉLETE 10. ELŐADÁS

Hány darabot adtak el 2002. jan. 15-én az A123 kódú áruból, mi a neve és az ára?

$$\{s^{(3)} \mid \exists u \exists v (\text{MENNYISÉG}(u) \wedge \text{ÁRU}(v) \wedge u[1] = 2002-01-15 \wedge u[2] = 'A123' \wedge$$
$$v[1] = 'A123' \wedge s[1] = u[3] \wedge s[2] = v[2] \wedge s[3] = v[3])\}$$
$$\{x, y, z \mid \exists u \exists v (\text{MENNYISÉG}(u, v, x) \wedge \text{ÁRU}(v, y, z) \wedge u = 2002-01-15 \wedge v = 'A123')\}$$

Mely nevű áruk azok, amelyekkel van azonos egységárú másik áru? $\{s^{(1)} \mid$

$$\exists u \exists v (\text{ÁRU}(v) \wedge \text{ÁRU}(u) \wedge s[1] = v[2] \wedge v[3] = u[3] \wedge \neg(v[1] = u[1]))\}$$
$$\{v \mid \exists x \exists y \exists w \exists u (\text{ÁRU}(x, v, u) \wedge \text{ÁRU}(y, w, u) \wedge x \neq y)\}$$

ADATBÁZISOK ELMÉLETE 10. ELŐADÁS

Példák oszlopalkulás alkalmazására

ÁRU(ÁRUKÓD, ÁRUNÉV, EGYSÉGÁR)
MENNYISÉG(DÁTUM, ÁRUKÓD, DB)
BEVÉTEL(DÁTUM, ÖSSZEG)
BEFIZ(ÖSSZEG, BEFIZ) BEFIZ=ÖSSZEG-4000

Az 2002. jan. 1. utáni napok bevételei a dátummal együtt, legyen előbb a dátum:

$$\{t^{(2)} \mid \text{BEVÉTEL}(t) \wedge t[1] \geq 2002-01-01\}$$
$$\{y, x \mid \text{BEVÉTEL}(y, x) \wedge y \geq 2002-01-01\} \quad (\text{sajtőhiba a Gajdos könyvben})$$

Az 2002. jan. 15-i bevétel és a befizetett összeg:

$$\{u^{(2)} \mid \text{BEFIZ}(u) \wedge \exists v (\text{BEVÉTEL}(v) \wedge v[1] = 2002-01-15 \wedge v[2] = u[1])\}$$
$$\{x, y \mid \text{BEFIZ}(x, y) \wedge \exists z (\text{BEVÉTEL}(z, x) \wedge z = 2002-01-15)\}$$

ADATBÁZISOK ELMÉLETE 10. ELŐADÁS

Biztonságos kifejezés

Dom(ψ) és a biztonságos formula és kifejezés ugyanaz, mint sorkalkulusnál.

Ugyanolyan technikák vannak a biztonságosság elérésére, mint sorkalkulusnál.

Tétel. A sorkalkulus és az oszlopalkulás ekvivalensek. A biztonságos sorkalkulus és a biztonságos oszlopalkulás ekvivalensek.

Bizonyítás: Vázlat:

$$\{t^{(k)} \mid \phi(t^{(k)})\} \longleftrightarrow \{u_1, \dots, u_k \mid \psi(u_1, \dots, u_k)\}$$
$$t^{(k)} \longleftrightarrow u_1, \dots, u_k$$
$$R(t^{(k)}) \longleftrightarrow R(u_1, \dots, u_k)$$
$$t^{(k)}[j] \longleftrightarrow u_j$$
$$\exists t^{(k)} \phi(t^{(k)}) \longleftrightarrow \exists u_1 \dots \exists u_k \psi(u_1, \dots, u_k)$$
$$\text{biztonságos} \longleftrightarrow \text{biztonságos}$$

Lekérdezőnyelvek típusai, általános jellemzőik

- Lehetnek algebrai alapúak: relációs algebrán alapuló lekérdezés, procedurális leírás
Előny: lekérdezésoptimalizálásra jobb
- Lehetnek logikai alapúak: sor vagy oszlopkalkulusra épülő lekérdezés, deklaratív leírás
Előny: könnyebben átlátható

Általában a konkrét lekérdezőnyelvek eltérnek a modelltől (algebrai és logikai esetben is), van amiben többet tudnak, van amiben kevesebbet, vagy csak máshogy

Lehetséges eltérések:

- logikai alapúakban eleve csak biztonságos kifejezéseknek megfelelő kérdéseket lehet írni (nincsenek kvantorok)

4

Példák relációs adatbáziskezelő nyelvekre

- **Information System Base Language**
 - relációs algebra alapú
 - kifejlesztő: IBM's United Kingdom Scientific Center
- **Query-By-Example**
 - oszlopkalkulus alapú
 - kifejlesztő: IBM's Watson Research Center
- **Structured Query Language**
 - oszlopkalkulus-szerű alapjai vannak, némi sorkalkulus elemmel
 - kifejlesztő: IBM's San Jose Research Laboratory

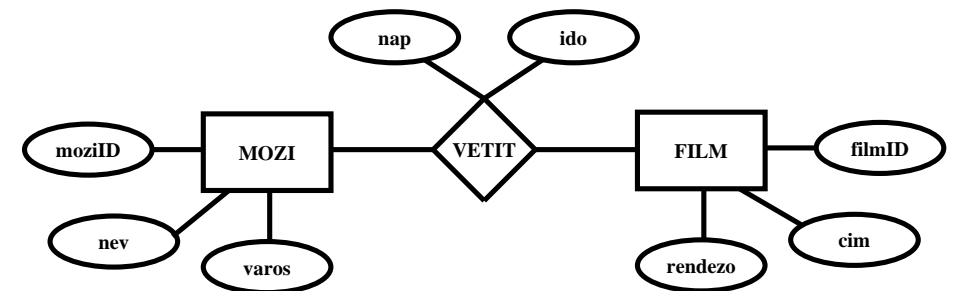
6

- aritmetika
- aggregátumok
- többszörös sorok kezelése/megengedése
- attribútumok sorrendje kötött

De az igaz mindre, hogy relációsan teljeseek (esetleg bizonyos műveletek nehezebben mennek) és általában van a lekérdező funkció mellett DML és DDL is.

5

A példákban használt relációs séma



7

MOZI	moziID	nev	varos
	101	Művész	Budapest
	102	Uránia	Pécs
	⋮		

FILM	filmID	cim	rendezo
	1	Macskajaj	E. Kusturica
	2	Moszkva tér	Török F.
	⋮		

VETIT	moziID	filmID	nap	ido
	101	1	péntek	16:00
	101	2	szombat	19:00
	⋮			

- adatfrissítő műveletek
- átnevezés
- kimenet formázása

ISBL

- Relációs algebra alapú lekérdezések
 - operátorok

művelet	relációs algebra	ISBL
unió	$R \cup S$	R + S
metszet	$R \cap S$	R . S
természetes illesztés	$R \bowtie S$	R * S
különbség (általánosabb)	$R \setminus (R \bowtie S)$	R - S
kiválasztás	$\sigma_F(R)$	R:F
vetítés	$\pi_{A,B,C}(R)$	R % A,B,C

- egyéb elemek
 - * eredmény megjelenítése: **list** kulcsszó
 - * értékadás relációinak: =
 - * és, vagy, tagadás jelei: &, |, ¬
- További nyelvi elemek
 - aggregátumok (min., max., összeg, átlag, darabszám) kezelése

ISBL példák

- Budapesti mozik nevei
list MOZI : varos = "Budapest" % nev
- Pénteken 16 órakor kezdődő filmek címe, rendezője
list FILM * VETIT : nap = "péntek" & ido = "16:00" % cim, rendezo
- Pénteken nem vetített filmek címe
list FILM - (VETIT : nap = "péntek") % cim
(Figyelem! Különbségnek más a definíciója.)
- Pénteken és szombaton is vetített filmek címe
list (FILM * VETIT : nap = "péntek" % cim) .
(FILM * VETIT : nap = "szombat" % cim)
vagy köztes relációk bevezetésével:
r1 = FILM * VETIT : nap = "péntek" % cim
r2 = FILM * VETIT : nap = "szombat" % cim
list r1.r2

QBE

- Oszlopkalkulus alapú lekérdezések, kétdimenziós
 - A lekérdezés elemei **változókkal** és **konstansokkal** kitöltött **sablon(ok)**
 - * jelölések

változó	aláhúzott változónév
konstans	nem aláhúzott érték
egyszer említett változó	üres cella
kimenetre kerülő attribútum	P. prefix

- * Példa: Budapesti mozik nevei

MOZI	mozilID	nev	varos
	<u>1</u>	P.mozinev	Budapest

vagy

MOZI	mozilID	nev	varos
		P.	Budapest

12

- aggregátumok kezelése
- reláció tranzitív lezártjának kezelése
- adatmódosító műveletek
- típusdefiníció, sémalétrehozás

Megjegyzés: dupla példányt kiirtja, azaz többszörös sorok nincsenek

14

- Összetett lekérdezések** is lehetségesek (használatukkor az **azonos nevű változók** illesztése történik meg).
 - * használható több soros sablon (ekkor a kiértékeléskor mindegyik sornak egy-egy futó sorváltozó fog megfelelni, és ha illeszkedés van, akkor megtörténik a kiírás)
 - * használható több sablon (kiértékelés hasonlóan, mint a többsoros kérdésnél, csak a sorváltozók nem ugyanazon reláció sorait futják be)
- A kiválasztás feltételeinek megadása
 - * Az **egyenlőség** konstanshoz való illesztéssel vizsgálható (mint az előbb),
 - * **egyéb egyszerű relációkhoz** a $\neq, >, <, >=, <=$ operátorok használhatók,
 - * **összetett feltételeket** (pl. két változó közt a $<$ relációt) külön feltételsablon megadásával lehet vizsgálni.
- További nyelvi elemek
 - mintaillesztés
 - aritmetika
 - kimenet rendezése
 - csoportosítás

13

QBE példák még

- Nem budapesti mozik nevei

MOZI	mozilID	nev	varos
		P.	\neq Budapest

- Pénteki és szombati kezdési időpontok

VETIT	mozilID	filmID	nap	ido
			P.péntek	P.
			P.szombat	P.

- Időpontok, amikor pénteken és szombaton is kezdődik film

VETIT	mozilID	filmID	nap	ido
			péntek	P.kezdes
			szombat	kezdes

15

- Pénteken vetített filmek adatai

FILM	filmID	cim	rendezo
	<u>1</u>	P.	P.

VETIT	mozilD	filmID	nap	ido
		<u>1</u>	péntek	

- Azok a városok, ahol van legalább két mozi:

MOZI	mozilD	nev	varos
	<u>1</u>		P.városnév
	<u>2</u>		városnév

CONDITIONS
<u>1</u> ≠ <u>2</u>

- SQL99 (SQL3, ebből is pár dolog, pl. trigger, rekurzió)

Működő rendszerekben ezek verziói vannak (főleg SQL2)

Az SQL nyelv

- Relációs nyelv, mint az eddigiek
- oszlopkalkulus jellegű, de némi sorkalkulusos beütéssel

Termékek, (amik szükségszerűen relációs nyelvet is tartalmaztak):

- IBM: System/R
- Relational Software: Oracle
- Relational Systems: Ingres

Szabványok

- SQL89 (SQL1)
- SQL92 (SQL2, mi nagyrészt ezt nézzük most)

Fontosabb utasítások

Data Definition Language:

- CREATE - séma létrehozása
- ALTER - séma módosítása
- DROP - séma törlése

Data Modification Language:

- INSERT - adatok beszúrása
- UPDATE - adatok módosítása
- DELETE - adatok törlése
- SELECT - adatok lekérdezése

Természetesen előbb mindig a sémát kell létrehozni, és utána dolgozhatunk vele, de most fordítva tárgyaljuk mert eddig a lekérdező nyelvekről volt szó.