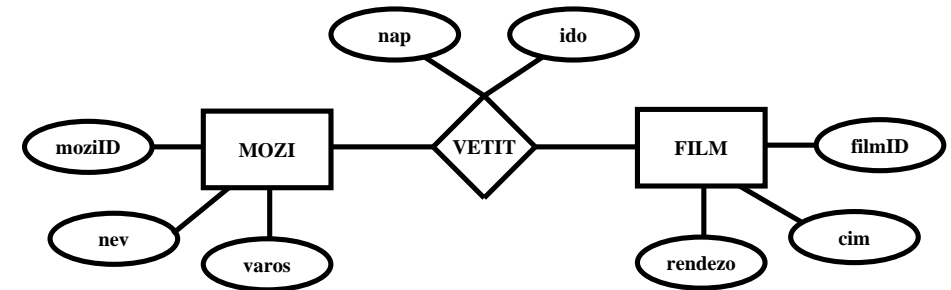


Adatbázisok elmélete 12. előadás

Csima Judit — Katona Gyula Y.
Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudományi Tsz.
I. B. 137/b
{csima,kiskat}@cs.bme.hu

2002. Október 25.

A példákban használt relációs séma



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
- **QUERY Language**
 - sorkalkulus alapú
 - kifejlesztő: Berkeley Egyetem
- **Structured Query Language**
 - oszlopkulus-szerű alapjai vannak, némi sorkalkulus elemmel
 - kifejlesztő: IBM's San Jose Research Laboratory

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
	⋮			

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)	$R \setminus (R \times S)$	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: **=**
- További nyelvi elemek
 - aggregátumok (min., max., összeg, átlag, darabszám) kezelése
 - adatfrissítő műveletek

4

QBE

- Oszlopkalkulus alapú lekérdezések
 - 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

FILM	filmID	cim	rendezo
	<u>1</u>	filmcim	E. Kusturica

- **Összetett lekérdezések** is lehetségesek (használatukkor az **azonos nevű változó**k illesztése történik meg).
 - * használható több soros sablon
 - * használható több sablon

6

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

5

- A kiválasztás feltételeinek megadása
 - * Az **egyenlőség** konstanshoz való illesztéssel vizsgálható,
 - * **egyéb egyszerű relációkhoz** a **\neg , =, >, <, >=, <=** operátorok használhatók,
 - * **összetett feltételeket** (pl. két változó közt a < relációt) külön feltételsablon megadásával lehet vizsgálni.
- További nyelvi elemek
 - csoportosítás
 - aggregátumok kezelése
 - reláció tranzitív lezártjának kezelése
 - adatmódosító műveletek

7

QBE példák

- Budapesti mozik nevei

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

vagy

MOZI	mozilD	nev	varos
		P.	Budapest

- Nem budapesti mozik nevei

MOZI	mozilD	nev	varos
		P.	≠ Budapest

Az SQL nyelv

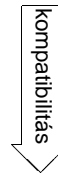
1970 E. F. Codd: a relációs adatmodell — ehhez egy nyelvre is szükség volt. Láttunk már példákat (ISBL, QUEL, QBE).

Termékek, (amik szükségszerűen relációs nyelvet is tartalmaztak):

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

Szabványok (ISO/ANSI):

- SQL89 (SQL1)
- SQL92 (SQL2, mi nagyrészt ezt nézzük most)
- SQL99 (SQL3)



- Pénteki és szombati kezdési időpontok

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

- Időpontok, amikor pénteken és szombaton is kezdődik film

VETIT	mozilD	filmID	nap	ido
			P.péntek	P.kezdes
			P.szombat	P.kezdes

- Pénteken vetített filmek adatai

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

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

Fontosabb utasítások

Data Modification Language:

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

Data Definition Language:

- CREATE - objektum létrehozása
- DROP - objektum törlése

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

DML utasítások — SELECT

Ezzel valósítható meg a kiválasztás, vetítés és a szorzat.

Szintaxis: **SELECT** <reláció>.<attrib₁>, ..., <reláció_j>.<attrib_n>
FROM <reláció₁>, ..., <reláció_m>
WHERE <kifejezés>

Relációs algebrabeli megfelelője:

$$\pi_{\langle \text{attrib}_1, \dots, \text{attrib}_n \rangle} \sigma_{\langle \text{kifejezés} \rangle} (\langle \text{reláció}_1 \rangle \times \dots \times \langle \text{reláció}_m \rangle)$$

Példa 1: A budapesti mozik azonosítói és nevei

SELECT mozi.mozilID, mozi.nev **FROM** mozi **WHERE** mozi.varos="Budapest"

Példa 2: A pénteken hétkor kezdődő filmek azonosítói

SELECT vetit.filmID **FROM** vetit **WHERE** vetit.nap="péntek" **AND** vetit.ido="19:00"

Megjegyzés: A szorzatra csak a sorváltozók bevezetése után tudunk példát nézni.

12

Attribútumhivatkozások

Amikor egy attribútumra akarunk hivatkozni, három lehetőségünk van:

- <attribútum> (ha ez egyértelmű)
- <reláció>.<attribútum> (ha ez egyértelmű – N.B.: egy reláció többször is szerepelhet a FROM után, lesz példa)
- <sorváltozó>.<attribútum> (mindig használható, ez tulajdonképpen egy sorváltozó)

Példa 3: A pénteken vetített filmek címei és rendezői (természetes illesztés)

SELECT cim, rendezo **FROM** film, vetit **WHERE** vetit.filmID = film.filmID **AND** nap="péntek"

Példa 4: Azok a várospárok, ahol vannak azonos nevű mozik (**FONTOS**)

SELECT m1.varos, m2.varos **FROM** mozi **AS** m1, mozi **AS** m2 **WHERE** m1.nev = m2.nev **AND** m1.varos <> m2.varos

Megjegyzés: a várospárok mindkét sorrendben megjelennek, és több azonos nevű mozi esetén többször is megjelennek.

14

SQL Sor- és oszlopváltozók

A FROM után felsorolt relációkhoz **sorváltozókat** rendelhetünk. Szintaxis (FROM után <reláció> helyén): <reláció> **AS** <sorváltozó> (Oracle-ben nem kell AS)

A SELECT után elhelyezett attribútum-hivatkozásokhoz **oszlopváltozókat** rendelhetünk.

Szintaxis (SELECT után <reláció>.<attrib> helyén):

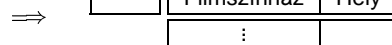
<reláció>.<attrib> **AS** <oszlopváltozó> (Oracle-ben AS opcionális)

A <reláció>. előtag elhagyható, ha egyértelmű, hogy melyik relációról van szó, továbbá a <reláció>. előtag helyett <sorváltozó>. előtag is szerepeltethető.

A sorváltozók valójában csak az eredményreláció attribútumainak elnevezésére használhatók, a SELECT utasításon belül nem hivatkozhatunk rájuk.

Például:

SELECT nev **AS** Filmszínház,
varos **AS** Hely **FROM** mozi



13

A WHERE kifejezés

Kifejezés felépítése:

- logikai műveletek: **AND, OR, NOT**
- összehasonlítás: **=, <, >, <=, LIKE, BETWEEN**
- aritmetikai műveletek: **+, -, *, DIV, MOD**
- karakterlánc műveletek: **CONCAT, SUBSTR, ...**
- változóhivatkozások: **<sorváltozó>.<attribútum>, <reláció>.<attribútum>, <attribútum>**
- konstans (szám,karakterlánc): **137, 42e-3, "fűzér"**
- NULL érték vizsgálata **IS NULL, IS NOT NULL** (később lesz)

15

LIKE és BETWEEN használata

LIKE használata:

- "_" egy tetszőleges karakterre illeszkedik
- "%" tetszőleges karakterláncra illeszkedik

BETWEEN használata: **BETWEEN a AND b** jelentése $a \leq . \leq b$

Példa 5: A 150 és 200 közötti azonosítójú mozik közül azok, amelyek B-vel kezdődő nevű városban vannak, és a nevük hárombetűs.

SELECT nev FROM mozi WHERE moziID BETWEEN 150 AND 200 AND varos LIKE "B%" AND nev LIKE "___"

Aggregátumok

Csoportosítsunk a varos attribútum szerint:

MOZI	moziID	nev	varos	szekszam
	1	Corvin	Budapest	2500
	4	Szindbád	Budapest	600
	5	Tabán	Budapest	200
	6	Uránia	Pécs	500
	2	Elit	Sopron	300
	3	Sopron Plaza Megaflex	Sopron	2000

Képezzük minden városra a székszámok összegét:

Aggregátumok

Lehetőségünk van bizonyos attribútumok értéke szerint csoportosítani az eredményt, és így aggregált sorokat képezni.

Például legyen adott a következő reláció:

MOZI	moziID	nev	varos	szekszam
	1	Corvin	Budapest	2500
	2	Elit	Sopron	300
	3	Sopron Plaza Megaflex	Sopron	2000
	4	Szindbád	Budapest	600
	5	Tabán	Budapest	200
	6	Uránia	Pécs	500

MOZI	varos	ossz_szekszam
	Budapest	3300
	Pécs	500
	Sopron	2300

Példa 6: Mindez SQL-ben

SELECT SUM(szekszam) AS ossz_szekszam FROM mozi GROUP BY varos

Aggregátumok

Amennyiben aggregátumokat képzünk a GROUP BY segítségével, akkor csak azokra az attribútumokra hivatkozhatunk közvetlenül a SELECT-ben, ami szerint csoportosítottunk. Ezen attribútumok értékei ugyanis egy aggregátumon belül jól meghatározottak. A többi attribútum az aggregátumon belül többféle értéket is felvehet. Ezért rájuk csak oszlopfüggvényeken keresztül hivatkozhatunk:

- SUM, AVG - összeg, átlag
- MIN, MAX - szélsőértékek
- COUNT - számosság

Példa 7: Az egyes városok legkisebb és legnagyobb mozijának mérete
SELECT varos, MIN(szekszam), MAX(szekszam) **FROM** mozi **GROUP BY** varos

Amennyiben a GROUP BY részt elhagyjuk, az összes sor egyetlen aggregált sorrá

DML utasítások — INSERT

Sorokat a relációba az INSERT utasítással szűrhetünk be.

Szintaxis: **INSERT INTO** <reláció> (<attrib₁>, ..., <attrib_n>) **VALUES** (<érték₁>, ..., <érték_n>)

Hatása: a <reláció> relációba egy új sor kerül, amiben <attrib₁> attribútum értéke <érték₁>, stb. A nem meghatározott értékű attribútumok a reláció létrehozásakor az attribútumhoz rendelt alapértelmezett értéket veszik fel.

Példa 9: Egy új film felvétele
INSERT INTO film (cim, rendezo) **VALUES** ("Egy csodálatos elme", "Ron Howard")

Megjegyzés: a filmID mező az alapértelmezett értékét kapja, de egy trigger (később lesz) segítségével akár automatikusan növekvő számozást is létrehozhatunk.

lesz összevonva, és az attribútumok értékeit csak oszlopfüggvényeken keresztül érhetjük el.

Példa 8: Az összes székszám
SELECT SUM(szekszam) **FROM** mozi

DML utasítások — UPDATE

Sorokat a relációban az UPDATE utasítással módosíthatunk.

Szintaxis: **UPDATE** <reláció> **SET** <attrib₁>=<érték₁>, ..., <attrib_n>=<érték_n>
WHERE <feltétel>

Hatása: a <reláció> reláció minden sorában, amelyik illeszkedik a <feltétel> feltételre <attrib₁> értéke <érték₁> lesz.

Példa 10: Az előbb beszúrt rendező nevének átírása rövidített alakba
UPDATE film **SET** rendezo="R. Howard" **WHERE** rendezo="Ron Howard"

DML utasítások — DELETE

Sorokat egy relációból a DELETE utasítással törölhetünk.

Szintaxis: **DELETE FROM** <reláció> **WHERE** <feltétel>

Hatása: a <reláció> reláció feltételre illeszkedő sorait törli.

Megjegyzés: a WHERE <feltétel> rész elhagyása esetén a reláció összes sorát törli.

Példa 11: Azon filmek törlése, amiknek a rendezője E. K. monogrammú
DELETE FROM film WHERE rendezo LIKE "E% K%"