



Koncepcija relacionog modela podataka

*Motivacija nastanka relacionog
modela podataka*

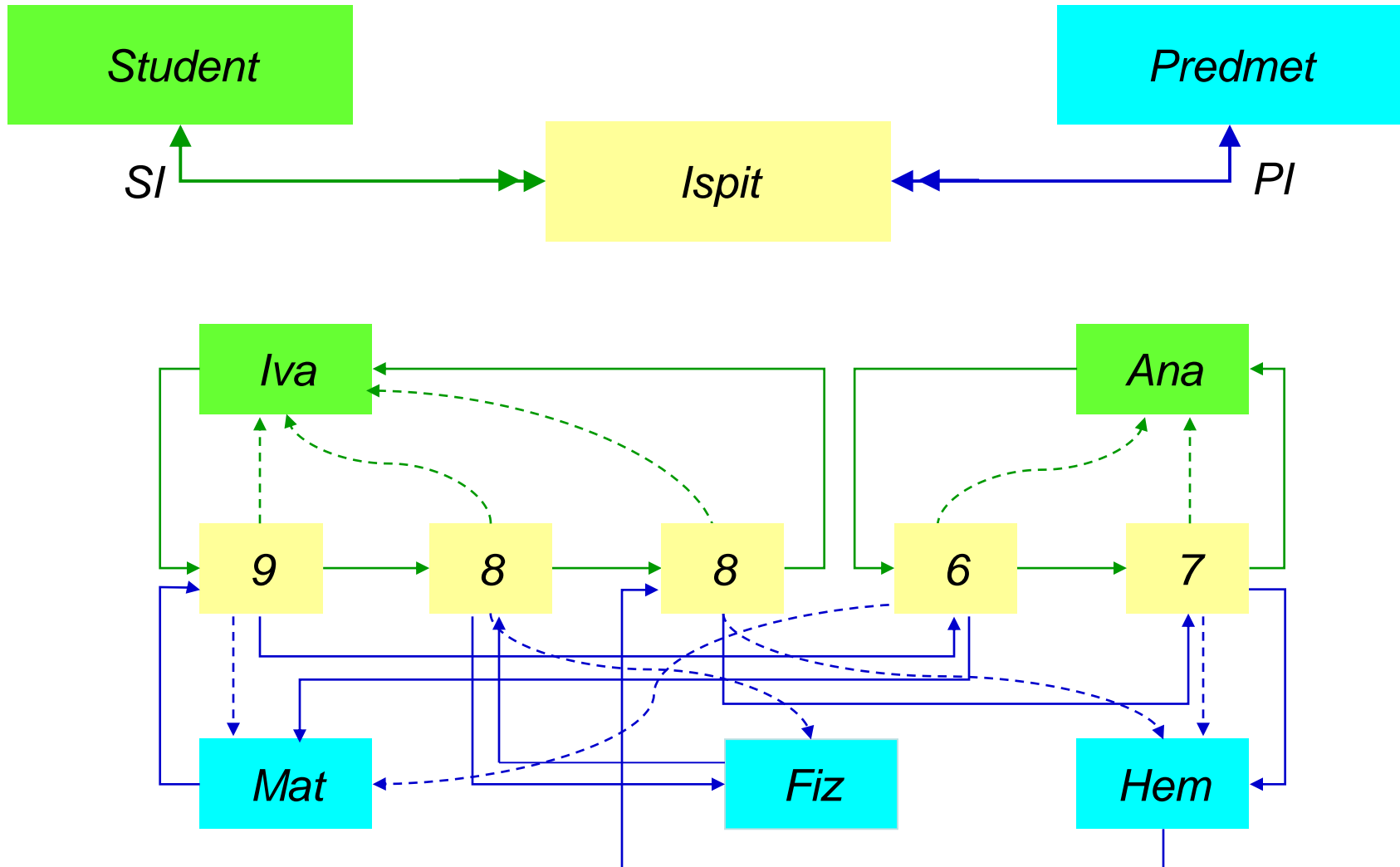
Sadržaj

- Mrežni i hijerarhijski MP
- Relacioni MP
- 12 principa RMP

Mrežni i hijerarhijski model

- U korišćenju 1970-ih godina
- Nedostaci
 - čvrsta povezanost programa i fizičke strukture podataka
 - strukturalna kompleksnost
 - proceduralno orijentisani jezici za manipulaciju podacima
 - "ad hoc" razvijeni modeli
 - bez značajnije upotrebe matematičkih formalizama

Mrežni i hijerarhijski model



Sadržaj

- Mrežni i hijerarhijski MP
- Relacioni MP
- 12 principa RMP

Relacioni model podataka

- U razvoju od 1970-ih godina
 - Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM* 13(6): 377-387.
 - Komercijalna upotreba od 199x-ih
- Motiv razvoja
 - otklanjanje nedostataka klasičnih modela podataka
 - čvrsta povezanost logičkih i fizičkih aspekata
 - strukturalna kompleksnost
 - navigacioni jezik
 - insistiranje na matematičkim osnovama u razvoju MP
 - matematička osnova RMP
 - teorija skupova i relacija
 - jako oslanjanje na predikatski račun I reda

Relacioni model podataka

- Zahtev
 - nezavisnost programa od podataka, tj. fizičke strukture podataka
- Kod ranijih MP
 - fizički aspekti BP ugrađeni u programe
 - raspodela slogova po zonama
 - fizički redosled \Rightarrow logičko grupisanje slogova
 - transformacija vrednosti ključa u adresu
 - lanci slogova sa pokazivačima
 - hijerarhijski redosled slogova
 - postupci zaštite podataka

Relacioni model podataka

- **Nezavisnost programa od podataka** - rešenje
 - potpuno odvajanje prezentacionog od formata memorisanja
 - relacija kao skup n-torki (torki)
 - apstraktni opis relacije: šema relacije

$$N(R, C)$$

- R - skup obeležja
- C - skup ograničenja, pri čemu je
 - $K \subseteq C$ – obavezno zadat skup ključeva, koji je neprazan
- često, u početnim fazama projektovanja, šema relacije se posmatra kao struktura

$$N(R, K)$$

Relacioni model podataka

- Primer
 - *Fakultet* ($\{SFK, NAZ, BIP\}, \{SFK\}$)
 - $r(Fakultet) = \{$
 - $(PMF, Matematički, 7),$
 - $(EKF, Ekonomski, 4),$
 - $(ETF, Elektrotehnički, 9),$
 - $(MAF, Mašinski, 7)$ $\}$
 - Upis nove torke
 - $(EKF, Elektronski, 8),$
 - narušio bi ograničenje ključa (uslov integriteta)

Relacioni model podataka

- **Strukturalna jednostavnost**
 - koncept relacije
 - osnova reprezentacije logičkih struktura podataka u RMP
 - ne sadrži nikakve informacije o fizičkoj organizaciji podataka
 - "prirodna" upotreba jednog od fundamentalnih matematičkih koncepata
 - relacija – "homogena" i "uniformna" struktura
 - lako razumljiva korisnicima podataka
 - uobičajena je tabelarna reprezentacija
 - » šema relacije \Rightarrow opis tabele (definicija tabele)
 - » relacija \Rightarrow kompletan sadržaj tabele (tabela)

Relacioni model podataka

- **Strukturalna jednostavnost**
 - selekcija podataka u operacijama nad BP
 - kod ranijih MP
 - upotreba fizičkih (relativnih ili apsolutnih) adresa
 - » pozicioniranje upotrebom indikatora aktuelnosti ili
 - » pozicioniranje putem odnosa između podataka
 - kod relacionog MP
 - asocijativno adresiranje
 - » isključiva upotreba simboličkih adresa – vrlo često vrednosti ključa
 - » svaki podatak u BP pronalazi se na osnovu naziva relacije, zadatih obeležja i vrednosti ključa
 - » skup n-torki (torki) sa zajedničkom osobinom selektira se na uniforman način – zadavanjem istog logičkog uslova
 - » SUBP vodi računa o transformaciji simboličke u relativnu adresu

Relacioni model podataka

- Primer

Fakultet

SFK	NAZ	BIP
<i>FIL</i>	<i>Filozofski</i>	<i>1</i>
<i>PMF</i>	<i>Matematički</i>	<i>7</i>
<i>ETF</i>	<i>Elektrotehnički</i>	<i>9</i>
<i>EKF</i>	<i>Ekonomski</i>	<i>4</i>
<i>MAF</i>	<i>Mašinski</i>	<i>7</i>

Projektant

MBR	IME	PRZ	SFK
<i>M3</i>	<i>Iva</i>	<i>Ban</i>	<i>PMF</i>
<i>M1</i>	<i>Ana</i>	<i>Tot</i>	<i>MAF</i>
<i>M4</i>	<i>Ana</i>	<i>Ras</i>	<i>FIL</i>
<i>M8</i>	<i>Aca</i>	<i>Pap</i>	<i>ETF</i>
<i>M6</i>	<i>Iva</i>	<i>Ban</i>	<i>EKF</i>
<i>M5</i>	<i>Eva</i>	<i>Tot</i>	<i>ETF</i>

Relacioni model podataka

- **Strukturalna jednostavnost**
 - povezivanje podataka
 - kod ranijih MP
 - upotreba fizičkih (relativnih ili apsolutnih) adresa u funkciji pokazivača
 - fizičko pozicioniranje logički susednih podataka
 - » o čemu je svaki transakcioni program morao voditi računa
 - kod relacionog MP
 - upotreba simboličkih adresa – prenetih vrednosti ključa
 - rešenje putem prostiranja ključa
 - » uvođenje pojma stranog ključa i ograničenja referencijalnog integriteta
 - rešenje putem kreiranja posebne tabele sa prostiranjem ključeva
 - » u oba slučaja, transakcioni program ne vodi računa o pretvaranju simboličke u relativnu adresu

Relacioni model podataka

- **Deklarativni jezik**

- temelji se na primenjenim tehnikama povezivanja podataka sa prostiranjem ključa
- dva alata za upitni jezik
 - relacionala algebra
 - » definisana na osnovama teorije skupova i skupovnih operacija
 - skupovni operatori: unija, presek, razlika
 - specijalizovani skupovni operatori: spoj (join), projekcija, selekcija, itd.
 - relacioni račun
 - relacioni račun nad torkama
 - relacioni račun nad domenima
 - » definisani na osnovama predikatskog računa I reda

Relacioni model podataka

- **SQL - Structured Query Language**
 - zasnovan na relacionom računu nad torkama
 - deklarativan
 - rad sa skupovima podataka (torki)
 - osnovni oblik naredbe za upite SQL-a je

SELECT	<i><lista obeležja></i>
FROM	<i><lista relacija></i>
WHERE	<i><logički izraz></i>

Relacioni model podataka

- Primer
 - ***SELECT*** *IME, PRZ, BIP*
FROM *Fakultet, Projektant*
WHERE *BIP > 5 AND*
Fakultet.SFK = Projektant.SFK

<i>IME</i>	<i>PRZ</i>	<i>BIP</i>
<i>Iva</i>	<i>Ban</i>	<i>7</i>
<i>Ana</i>	<i>Tot</i>	<i>7</i>
<i>Aca</i>	<i>Pap</i>	<i>9</i>
<i>Eva</i>	<i>Tot</i>	<i>9</i>

Relacioni model podataka

- Primer
 - ***SELECT*** *IME, PRZ, BIP*
FROM *Fakultet* **NATURAL JOIN** *Projektant*
WHERE *BIP > 5*

<i>IME</i>	<i>PRZ</i>	<i>BIP</i>
<i>Iva</i>	<i>Ban</i>	<i>7</i>
<i>Ana</i>	<i>Tot</i>	<i>7</i>
<i>Aca</i>	<i>Pap</i>	<i>9</i>
<i>Eva</i>	<i>Tot</i>	<i>9</i>

Sadržaj

- Mrežni i hijerarhijski MP
- Relacioni MP
- 12 principa RMP

12 principa RMP

- Codd, E. F. (1990).
 - "The Relational Model for Data Base Management: Version 2", Addison Wesley.
 - **Rule 0:** The system must qualify as relational, as a database, and as a management system
 - For a system to qualify as a relational database management system (RDBMS), that system must use its *relational* facilities (exclusively) to *manage* the database.
 - **Rule 1:** The *information rule*
 - All information in the database is to be represented in one and only one way, namely by values in column positions within rows of tables.

12 principa RMP

– Rule 2: The *guaranteed access rule*

- All data must be accessible with no ambiguity. This rule is essentially a restatement of the fundamental requirement for primary keys. It says that every individual scalar value in the database must be logically addressable by specifying the name of the containing table, the name of the containing column and the primary key value of the containing row.

– Rule 3: *Systematic treatment of null values*

- The DBMS must allow each field to remain null (or empty). Specifically, it must support a representation of "missing information and inapplicable information" that is systematic, distinct from all regular values (for example, "distinct from zero or any other number", in the case of numeric values), and independent of data type. It is also implied that such representations must be manipulated by the DBMS in a systematic way.

12 principa RMP

- **Rule 4:** *Active online catalog based on the relational model*
 - The system must support an online, inline, relational catalog that is accessible to authorized users by means of their regular query language. That is, users must be able to access the database's structure (catalog) using the same query language that they use to access the database's data.

12 principa RMP

- **Rule 5:** The *comprehensive data sublanguage rule*
 - The system must support at least one relational language that
 - » Has a linear syntax (whose statements are expressible, per some well-defined syntax, as character strings)
 - » Can be used both interactively and within application programs,
 - » Supports data definition operations (including view definitions), data manipulation operations (update as well as retrieval), security and integrity constraints, and transaction management operations (begin, commit, and rollback).

- **Rule 6:** The *view updating rule*
 - All views that are theoretically updatable must be updatable by the system.

12 principa RMP

- **Rule 7:** *High-level insert, update, and delete*
 - The system must support set-at-a-time *insert*, *update*, and *delete* operators. This means that data can be retrieved from a relational database in sets constructed of data from multiple rows and/or multiple tables. This rule states that insert, update, and delete operations should be supported for any retrievable set rather than just for a single row in a single table.

- **Rule 8:** *Physical data independence*
 - Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure.

12 principa RMP

– Rule 9: *Logical data independence*

- Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure. Logical data independence is more difficult to achieve than physical data independence.

– Rule 10: *Integrity independence*

- Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications.

12 principa RMP

– Rule 11: *Distribution independence*

- The distribution of portions of the database to various locations should be invisible to users of the database. Existing applications should continue to operate successfully:
 - » when a distributed version of the DBMS is first introduced; and
 - » when existing distributed data are redistributed around the system.

– Rule 12: The *nonsubversion rule*

- If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).

Sadržaj

- Mrežni i hijerarhijski MP
- Relacioni MP
- 12 principa RMP

Literatura

- Pavle Mogin, Ivan Luković: Principi baza podataka
 - Poglavlje 4.1

Pitanja i komentari



Koncepcija relacionog modela podataka

*Motivacija nastanka relacionog
modela podataka*