Relacioni upitni jezici

- Praktični rezultat formalnih istraživanja RM
- Nadgradnja programskih jezika opšte namene
- Relaciona algebra
- Relacioni račun
- SQL-oliki
- Interaktivni / aplikativni

Structured Query Language - SQL

- IBM istraživačka laboratorija, San Jose:
- SQUARE (1973), SEQUEL (1974), SEQUEL 2 (1976), SQL
- Standardi: 1986, 1989, 1992, 1999, 2003, 2006, 2008, 2011.
  - DDL
  - DML
  - DCL

SQL - standardi

- 1986: tabelarni model podataka, definicija sheme, pogleda i kursora
- 1989: referencijalni integritet i ugnježdenje SQL-a u Ada, C, COBOL, FORTRAN, Pascal i PL/I.
- 1992: spoljašnje spajanje, kaskadno ažuriranje i brisanje, skupovne operacije, definicija domena, novi tipovi podataka, itd.
- 1999: regularni izrazi, rekurzivni upiti, trigeri, kontrolne strukture, novi tipovi, OO-svojstva
- 2003: svojstva vezana za XML
- 2006: integracija XML-a i XQuery-a u SQL
- 2008: najveće dopune u delu „Foundations“
- 2011: dalje dopune DML, DDL

Structured Query Language – SQL: DML

- **Primer**: naslovi romana u izdanju “Prosvete”

```
SELECT NASLOV
FROM K JOIN KI on K.K_SIF= KI.K_SIF
       JOIN I on KI.I_SIF=I.I_SIF
WHERE OBLAST = 'roman' AND NAZIV =
  'Prosveta'
```

SQL – DML

`SELECT` – pretraživanje

`INSERT` – unošenje

`UPDATE` – ažuriranje

`DELETE` – brisanje

(od standarda 2003):

`MERGE` – pripajanje

SQL: pretraživanje

Jednostavni upitni blok: osnovne linije

```
SELECT lista-kolona
FROM ime-tabele
WHERE logički-izraz
```

SQL: jednostavni upitni blok

```
SELECT I_SIF, STATUS
FROM I
WHERE DRZAVA = 'Amerika'
```

Rezultat:

LSIF	STATUS
i2	20

SQL: projekcija

```
SELECT K_SIF
FROM KI
```

Rezultat:

K_SIF
k1
k2
k3
k4
k5
k6
k6

SQL: projekcija sa eliminacijom duplikata

```
SELECT DISTINCT K_SIF
FROM KI
```

Rezultat:

K_SIF
k1
k2
k3
k4
k5
k6

SQL: cela tabela

```
SELECT *
FROM I
```

isto što i

```
SELECT I_SIF, NAZIV, STATUS, DRZAVA
FROM I
```

SQL: skalarni izraz u SELECT liniji

```
SELECT K_SIF,I_SIF, IZDANJE,
       'Tiraz u hijadama=', TIRAZ/1000
FROM KI
```

K_SIF	I_SIF	IZDANJE	4	5
k1	i1	2	Tiraz u hiljadama=	10
k2	i1	2	Tiraz u hiljadama=	7
k3	i1	1	Tiraz u hiljadama=	10
k4	i1	2	Tiraz u hiljadama=	10
k5	i2	4	Tiraz u hiljadama=	5
k6	i3	1	Tiraz u hiljadama=	3
k6	i4	3	Tiraz u hiljadama=	5

SQL: logički izraz u WHERE liniji

```
SELECT I_SIF
FROM I
WHERE DRZAVA='Srbija'
 AND STATUS > 20
```

Rezultat:

I_SIF
i1
i4

SQL: BETWEEN u logičkom izrazu

```
SELECT I_SIF,
 STATUS
FROM I
WHERE STATUS
 BETWEEN 15 AND 30
```

Rezultat:

I_SIF	STATUS
i1	30
i2	20
i4	30

SQL: IN u logičkom izrazu

```
SELECT I_SIF,
 STATUS
FROM I
WHERE STATUS IN
 (20, 30)
```

Rezultat:

I_SIF	STATUS
i1	30
i2	20
i4	30

SQL: LIKE u logičkom izrazu

```
% 0 ili više
 karaktera,
_ 1 karakter
```

```
SELECT I_SIF, DRZAVA
FROM I
WHERE DRZAVA LIKE
 '%j%'
```

Rezultat:

I_SIF	DRZAVA
i1	Srbija
i3	Srbija
i4	Srbija

SQL: IS NULL u logičkom izrazu

```
SELECT I_SIF, IZDANJE
FROM KI
WHERE K_SIF=‘k1’ AND
 TIRAZ IS NOT NULL
```

I_SIF	IZDANJE
i1	2

SQL: upiti spajanja – primer dve tabele

```
SELECT *
FROM I JOIN P on I.DRZAVA = P.DRZAVA
```

isto što i

```
SELECT I.*, P.*
FROM I JOIN P on I.DRZAVA = P.DRZAVA
```

SQL: primer upita spajanja - rezultat

I_SIF	NAZIV	ST.	DRZ.	P_SIF	IME	BR_N.	DRZ.
i1	Prosveta	30	Srbija	p1	B.Ćopić	2	Srbija
i1	Prosveta	30	Srbija	p3	B.Šimšić	1	Srbija
i1	Prosveta	30	Srbija	p4	D.Maksimović	2	Srbija
i2	Add.Wesley	20	Amer.	p5	C.J.Date	1	Amer.
i3	D.novine	10	Srbija	p1	B.Ćopić	2	Srbija
i3	D.novine	10	Srbija	p3	B.Šimšić	1	Srbija
i3	D.novine	10	Srbija	p4	D.Maksimović	2	Srbija
i4	M.srpska	30	Srbija	p1	B.Ćopić	2	Srbija
i4	M.srpska	30	Srbija	p3	B.Šimšić	1	Srbija
i4	M.srpska	30	Srbija	p4	D.Maksimović	2	Srbija

(skraćenja oznaka vrednosti su iz prostornih razloga)

SQL: primer upita spajanja - 2

```
SELECT I.I_SIF, P.P_SIF
FROM I JOIN P on I.DRZAVA = P.DRZAVA
WHERE I.STATUS > 20
```

SQL: primer upita spajanja – 2 - rezultat

I_SIF	P_SIF
i1	p1
i1	p3
i1	p4
i4	p1
i4	p3
i4	p4

SQL: spajanje n (n>2) tabela

```
SELECT DISTINCT I.DRZAVA AS IDRZAVA, P.DRZAVA AS
  PDRZAVA
FROM I JOIN KI on I.I_SIF = KI.I_SIF
       JOIN KP on KI.K_SIF = KP.K_SIF
       JOIN P on KP.P_SIF = P.P_SIF
```

Rezultat:

IDRZAVA	PDRZAVA
Srbija	Srbija
Srbija	Engleska
Amerika	Amerika

SQL: spajanje tabele sa samom sobom

Naći parove šifara izdavača iz iste države

(Naći parove šifara izdavača i pisaca iz iste države:

```
SELECT I.I_SIF, P.P_SIF
FROM I JOIN P on I.DRZAVA = P.DRZAVA)
```

```
SELECT PRVI.I_SIF, DRUGI.I_SIF
FROM I PRVI JOIN I DRUGI on
     PRVI.DRZAVA = DRUGI.DRZAVA
```

SQL: spajanje tabele sa samom sobom - rezultat

I_SIF	I_SIF
i1	i1
i1	i3
i1	i4
i2	i2
i3	i1
i3	i3
i3	i4
i4	i1
i4	i3
i4	i4

SQL: spajanje tabele sa samom sobom – verzija 2

```
SELECT PRVI.I_SIF, DRUGI.I_SIF
FROM I PRVI JOIN I DRUGI on
     PRVI.DRZAVA = DRUGI.DRZAVA
WHERE PRVI.I_SIF < DRUGI.I_SIF
```

SQL: spajanje tabele sa samom sobom – verzija 2 - rezultat

I_SIF	I_SIF
i1	i3
i1	i4
i3	i4

SQL: podupiti

- Jedno od osnovnih svojstava SQL-a
- Predstavljanje pojedinačne vrednosti – operacija poređenja
- Predstavljanje skupa vrednosti (EXISTS, IN, SOME, ANY, ALL)

SQL: podupiti - operacija poređenja

```
SELECT I_SIF
FROM I
WHERE DRZAVA=(SELECT DRZAVA
              FROM I
              WHERE I_SIF = 'i1')
```

SQL: podupiti - IN operator

```
SELECT I.NAZIV
FROM I
WHERE I.I_SIF IN
      (SELECT KI.I_SIF
       FROM KI
       WHERE KI.K_SIF = 'k6')
```

SQL: ekvivalentni upit spajanja

```
SELECT I.NAZIV
FROM I JOIN KI on I.I_SIF = KI.I_SIF
WHERE KI.K_SIF = 'k6'
```

SQL: podupiti – primer: rezultat

NAZIV
Dečje novine
Matica srpska

SQL: podupiti - kvantifikovano poređenje

```
SELECT NAZIV
FROM I
WHERE I_SIF = SOME
              (SELECT I_SIF
              FROM KI
              WHERE K_SIF = 'k6')
```

- **ANY, ALL**

SQL: korelisani podupit

```
SELECT I.NAZIV
FROM I
WHERE 'k6' IN
  (SELECT KI.K_SIF
   FROM KI
   WHERE KI.I_SIF = I.I_SIF)
```

SQL: korelisani podupit nad jednom tabelom

Šifre svih knjiga koje izdaje više od jednog izdavača:

```
SELECT DISTINCT KIX.K_SIF
FROM KI KIX
WHERE KIX.K_SIF IN
      (SELECT KIY.K_SIF
       FROM KI KIY
       WHERE KIY.I_SIF <> KIX.I_SIF)
```

SQL: podupiti
- egzistencijalni kvantor

```
SELECT I.NAZIV
FROM I
WHERE EXISTS
  (SELECT *
   FROM KI
   WHERE KI.I_SIF = I.I_SIF AND KI.K_SIF =
  'k6')
```

SQL: podupiti
- univerzalni kvantor

$$(\text{FORALL x})\text{p(x)} \equiv \text{NOT } (\text{EXISTS x})(\text{NOT}(\text{p(x)}))$$

- Naći nazive izdavača koji izdaju sve knjige Branka Ćopića

SQL: podupiti
- univerzalni kvantor

- Knjige Branka Ćopića:

```
CREATE VIEW  K_BC AS
       (SELECT *
        FROM   K
        WHERE  K_SIF IN
              (SELECT K_SIF
               FROM   KP
               WHERE  P_SIF =
                     (SELECT P_SIF
                      FROM   P
                      WHERE  IME = 'Branko Ćopić')))
```

SQL: podupiti
- univerzalni kvantor

- Nazivi izdavača koji su izdali sve knjige B.Ć.

```
SELECT I.NAZIV
FROM I
WHERE NOT EXISTS
     (SELECT *
      FROM K_BC
      WHERE NOT EXISTS
          (SELECT *
           FROM KI
           WHERE KI.I_SIF=I.I_SIF AND KI.K_SIF=K_BC.K_SIF))
```

Rezultat:

NAZIV
Prosveta

SQL: skalarne funkcije

- Konverzione, npr. (*dec*)DECIMAL(*num/string*), (*string*)DIGITS(*num*), (*float/double*)FLOAT(*num/string*), (*int*)INTEGER(*num/string*) (DB2)
- Opšte, npr. LENGTH() – dužina podataka u bajtovima
- Aritmetičke, npr. ABS(), TAN(), SQRT(), TRUNC()
- Obrada znakovnih podataka: SUBSTR(), POSSTR()
- Nad specifičnim tipovima, npr. TIME(), YEAR(), DATE(), DAY() nad tipovima TIME, DATE, TIMESTAMP

SQL: skalarne funkcije, CASE-WHEN izraz

- SELECT NAZIV,
  CASE SUBSTR(DRZAVA,1,1)
  WHEN ‘S' THEN ‘Serbia'
  WHEN 'A' THEN 'USA'
  WHEN 'E' THEN 'UK'
  END AS DRZAVA_ENG
  FROM I

(Alternativno: uslov u WHEN)

NAZIV	DRZAVA_ENG
Prosveta	Serbia
Addison Wesley	USA
Decje novine	Serbia
Matica Srpska	Serbia

SQL: agregatne funkcije

- `COUNT`
- `SUM`
- `AVG`
- `MAX`
- `MIN`
- `----------`
- **STDDEV(expr)**
- **VARIANCE(expr) (VAR, VAR_POP)**
- **COVAR(expr1,expr2)**
- **CORRELATION(expr1,expr2)**
- **...**

SQL: agregatne funkcije – count

```
SELECT COUNT (DISTINCT I_SIF)
FROM KI
(rezultat?)

SELECT COUNT(*)
FROM KI
WHERE KI.K_SIF = 'k6'
(rezultat?)
```

Rezultat nikada nije NULL

Tabela KI

KI

K_SIF	I_SIF	IZDANJE	GODINA	TIRAZ
k1	i1	2	1965	10000
k2	i1	2	1974	7000
k3	i1	1	1975	10000
k4	i1	2	1979	10000
k5	i2	4	1986	5000
k6	i3	1	1966	3000
k6	i4	3	1988	5000

Tabele P, I, K

P

P_SIF	IME	BR_NASLOVA	DRZAVA
p1	B.Ćopić	2	Srbija
p2	M.Benson	1	Engleska
p3	B.Šljivić-Šimšić	1	Srbija
p4	D.Maksimović	2	Srbija
p5	C.J.Date	1	Amerika

I

I_SIF	NAZIV	STATUS	DRZAVA
i1	Prosveta	30	Srbija
i2	Addison Wesley Publ. Comp.	20	Amerika
i3	Dečje novine	10	Srbija
i4	Matica srpska	30	Srbija

K

K_SIF	NASLOV	OBLAST
k1	Osma ofanziva	roman
k2	Nemam više vremena	poezija
k3	Pionirska trilogija	roman
k4	Srpskohrvatsko-engleski rečnik	leksikografija
k5	An Introduction to Database Systems	računarstvo
k6	Tražim pomilovanje	poezija

SQL: agregatne funkcije – sum, max

```
SELECT SUM(TIRAZ)
FROM KI
WHERE KI.K_SIF = 'k6'
```

- (rezultat?)

Agregatna funkcija u where liniji???

```
SELECT I.I_SIF
FROM I
WHERE STATUS <
        (SELECT MAX(STATUS)
         FROM I)
```

- (rezultat?)

- Rezultat je NULL za prazan skup vrednosti

SQL: agregatne funkcije – avg

```
SELECT IX.I_SIF, IX.STATUS, IX.DRZAVA
FROM I IX
WHERE IX.STATUS >=
      (SELECT AVG(STATUS)
       FROM I IY
       WHERE IY.DRZAVA = IX.DRZAVA)
```

SQL: agregatne funkcije – `avg` – rezultat

Rezultat:

I_SIF	STATUS	DRZAVA
i1	30	Srbija
i2	20	Amerika
i4	30	Srbija

SQL: GROUP BY

```
SELECT K_SIF,
  SUM(TIRAZ)
FROM KI
GROUP BY K_SIF
```

Rezultat:

K_SIF	2
k1	10000
k2	7000
k3	10000
k4	10000
k5	5000
k6	8000

SQL: HAVING

```
SELECT K_SIF
FROM KI
GROUP BY K_SIF
HAVING COUNT(*) > 1
```

Rezultat:

K_SIF
k6

SQL: UNION, INTERSECT, EXCEPT

```
SELECT K_SIF
FROM K
WHERE OBLAST = 'roman'
UNION
SELECT K_SIF
FROM KI
WHERE I_SIF = 'i2'
```

Rezultat:

K_SIF
k1
k3
k5

SQL: tabela K

K

K_SIF	NASLOV	OBLAST
k1	Osma ofanziva	roman
k2	Nemam više vremena	poezija
k3	Pionirska trilogija	roman
k4	Srpskohrvatsko-engleski rečnik	leksikografija
k5	An Introduction to Database Systems	računarstvo
k6	Tražim pomilovanje	poezija

SQL: UNION, INTERSECT, EXCEPT

```
SELECT DRZAVA
FROM I
INTERSECT
SELECT DRZAVA
FROM P
```

Rezultat:

DRZAVA
Amerika
Srbija

SQL: UNION, INTERSECT, EXCEPT

```
SELECT DRZAVA
FROM I
INTERSECT ALL
SELECT DRZAVA
FROM P
```

Rezultat:

DRZAVA
Amerika
Srbija
Srbija
Srbija

SQL: UNION, INTERSECT, EXCEPT

```
SELECT DRZAVA
FROM P
EXCEPT
SELECT DRZAVA
FROM I
```

EXCEPT ALL???

Rezultat:

DRZAVA
Engleska

SQL: ORDER BY - 1

```
SELECT I_SIF,
 STATUS
FROM I
ORDER BY STATUS
 DESC
```

Rezultat:

LSIF	STATUS
i1	30
i4	30
i2	20
i3	10

SQL: ORDER BY - 2

```
SELECT K_SIF, I_SIF, IZDANJE, 'Tiraz u
 hiljadama =', TIRAZ/1000
FROM KI
ORDER BY 5, I_SIF
```

SQL: ORDER BY – 2
- rezultat

Rezultat:

K_SIF	L_SIF	IZDANJE	4	5
k6	i3	1	Tiraz u hiljadarna =	3
k5	i2	4	Tiraz u hiljadarna =	5
k6	i4	3	Tiraz u hiljadarna =	5
k2	i1	2	Tiraz u hiljadarna =	7
k1	i1	2	Tiraz u hiljadarna =	10
k3	i1	1	Tiraz u hiljadarna =	10
k4	i1	2	Tiraz u hiljadarna =	10

SQL: operator WITH

- MQT - materialized query table u db2 - izvedena imenovana tabela
- Iz tabele sa parovima (IME, NAZIV) koja sadrži imena pisaca i nazive izdavača koji izdaju knjige tih pisaca, izdati nazive svih izdavača Branka Ćopića.

- ```
  WITH PI AS
          (SELECT DISTINCT IME, NAZIV
           FROM P JOIN KP ON P.P SIF=KP.P SIF
                       JOIN  KI  ON KP.K SIF=KI.K SIF
                       JOIN  I    ON KI.I SIF=I.I SIF)
  SELECT NAZIV
  FROM PI
  WHERE IME='B.Copic'
  ```
  ```

SQL: operator WITH - rezultat

Rezultat:

NAZIV
Prosveta

SQL: Način izvršavanja SELECT iskaza

SELECT ...
FROM ...
WHERE ...
GROUP BY ...
HAVING ...
ORDER BY ...

1. Dekartov proizvod (FROM)
2. Restrikcija (WHERE)
3. Grupisanje (GROUP BY)
4. Izbor grupa (HAVING)
5. Projekcija (SELECT)
6. Uređenje (ORDER BY)

SQL: relaciona kompletnost

- 1. Unija, presek, razlika – direktno
- 2. Dekartov proizvod: FROM
- 3. Projekcija: SELECT
- 4. Restrikcija: WHERE
- 5. Spajanje: SELECT-FROM-WHERE
- 6. deljenje: *R*(*A;B*) : *S*(*C*)

```
SELECT A
FROM R RX
WHERE NOT EXISTS
        (SELECT *
         FROM S
         WHERE NOT EXISTS
                 (SELECT *
                  FROM R
                  WHERE R.B = S.C AND R.A = RX.A))
```